

DLR-IB-MO-HF-2020-235

**Funktionelle Nachbildung eines
Avionics Compartments für die
Forschung und Verifikation von
neuartigen Flugsystemen**

Bachelorarbeit

Ivan Belyaev



DLR

**Deutsches Zentrum
für Luft- und Raumfahrt**

BACHELORTHESIS

Ivan Belyaev

Funktionelle Nachbildung eines Avionics Compartments für die Forschung und Verifikation von neuartigen Flugsystemen

FAKULTÄT TECHNIK UND INFORMATIK

Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science

Department of Information and Electrical Engineering

Ivan Belyaev

Funktionelle Nachbildung eines Avionics
Compartments für die Forschung und
Verifikation von neuartigen Flugsystemen

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Jörg Dahlkemper
Zweitgutachter: Dipl.-Ing. Hendrik Meyer

Abgegeben am 26. Mai 2020

Ivan Belyaev

Thema der Bachelorthesis:

Funktionelle Nachbildung eines Avionics Compartments für die Forschung und Verifikation von neuartigen Flugsystemen

Stichworte

Avionics Compartment, Flugzeugreferenzarchitektur, ARINC 429, CAN-Bus, Robot Operating System, neuartige Flugzeugsysteme, Verifikation und Präsentation von Forschungsthemen, Open-Source

Kurzzusammenfassung

Die Bachelorarbeit beschäftigt sich mit der Entwicklung einer *Demonstratorplattform* für den Zweck der *Verifikation und Präsentation von neuartigen Forschungsthemen* aus dem Bereich der Luftfahrtinstandhaltung mit Schwerpunkt Kosten- und Entwicklungsaufwandminimierung: Es wird ein aus einem Gerüst, Abdeckungen, Stromverteilung sowie einem Halterungssystem bestehendes mechanisches Trägersystem für modulare Avionik-Komponenten konzipiert und erstellt.

In dieses System wird ein modularer, elektronischer Aufbau mit Grundbausteinen wie Raspberry PI's, Arduinos, Displays, Sensoren, CAN-USB Adapter sowie dem ROS-Framework integriert. In diesem Aufbau werden die luftfahrtspezifischen Komponenten *Electronic Instrument System*, *Air Data Inertial Reference Unit*, *Tire Pressure Indicating System* und *Braking and Steering Control Unit* modellhaft nachgebildet und zusammen mit der Demonstratorplattform aus Sicht eines Luftfahrtforschers verifiziert.

Title of the paper

Functional replication of an Avionics Compartment for the research and verification of new kinds of flight systems

Keywords

Avionics Compartment, Aircraft Reference Architecture, ARINC 429, CAN-Bus, Robot Operating System, novel Aircraft Systems, Verification and Presentation of Research Thesis, Open-Source

Abstract

The bachelor thesis deals with the development of a *demonstrator platform* for the purpose of *verification and presentation of novel research theses* in the field of aviation maintenance repair and operations. Its aim is to minimize costs and development efforts for new flight technologies. A mechanical support system for modular avionics components consisting of a framework, covers, a power distribution system and a mounting system is designed and built.

Subsequently, a modular, electronic installation with basic components such as Raspberry PI's, Arduinos, displays, sensors, CAN-USB adapters and the ROS framework is integrated into this platform. This installation is set up for modeling the aviation-specific units *Electronic Instrument System*, *Air Data Inertial Reference Unit*, *Tire Pressure Indicating System* and *Braking and Steering Control Unit* as well as for verifying the demonstrator platform from the perspective of an aviation researcher.

Inhaltsverzeichnis

Abkürzungsverzeichnis	iii
Abbildungsverzeichnis	iv
Tabellenverzeichnis	vi
1 Einleitung.....	1
1.1 Motivation	1
1.2 Ziel	1
1.3 Rahmen.....	2
1.4 Vorgehensweise	2
2 Ausgangssituation	3
2.1 Grundlagen.....	3
2.1.1 Verteilte Systeme	3
2.1.2 Eingebettete Systeme	4
2.1.3 Robot Operating System	5
2.2 Kommunikationsbusse in der Avionik	6
2.2.1 ARINC 429	6
2.2.2 CAN-Bus	8
2.3 Avionics Compartment	9
2.4 Electronic Instrument System	10
2.4.1 System Data Acquisition Computer	12
2.4.2 Display Management Computer	13
2.5 Weitere Flugzeugsysteme	14
2.5.1 Tire Pressure Indicating System	14
2.5.2 Brake and Steering Control Unit.....	14
2.5.3 Air Data Inertial Reference Unit.....	15
2.6 Zusammenhang zwischen PFD, SD, DMC, SDAC, ADIRU, TPIS und BSCU	16
3 Anforderungen	18
3.1 Anforderungen an den Demonstrator	18
3.1.1 Präsentierbarkeit	18
3.1.2 Erweiterbarkeit	18
3.1.3 Autarkie	19
3.1.4 Verfügbarkeit.....	20
3.1.5 Grenzmaße	20
3.1.6 Grenzgewicht	20
3.1.7 Personen- und Geräteschutz	21
3.2 Anforderungen an die Beispielanwendung	21
3.2.1 Identifizierung	21
3.2.2 Analyse	23
3.3 Tabellarische Zusammenfassung der Anforderungen	26
4 Konzeption	31
4.1 Mechanisches Trägersystem	31
4.1.1 Abdeckungen	32

4.1.2	Gerüst.....	32
4.1.3	Halterungssystem.....	34
4.2	Kommunikationssystem des Anwendungsfalls	34
4.2.1	Middleware.....	35
4.2.2	ARINC 429 Ersatz	36
4.3	Komponenten des Anwendungsfalls	36
4.3.1	Sensoren.....	36
4.3.2	Computer.....	39
4.3.3	Displays.....	42
4.4	Stromversorgungssystem	43
4.4.1	Anschluss.....	44
4.4.2	Wandlung.....	44
4.4.3	Verteilung.....	46
5	Entwicklung	48
5.1	Demonstrator	48
5.1.1	Mechanisches Trägersystem.....	48
5.1.2	Stromversorgungssystem	50
5.2	Die Beispielanwendung.....	52
5.2.1	Akquirierungseinheiten	53
5.2.2	Grundlegender Aufbau des verteilten Systems	57
5.2.3	Kommunikationsmanager für das verteilte System	59
5.2.4	Die Zusammenführung.....	63
6	Test	66
6.1	Beispielanwendung	66
6.1.1	Beispielmessung.....	66
6.1.2	Kommunikation	67
6.2	Demonstratortechnik	69
6.2.1	Präsentierbarkeit	69
6.2.2	Erweiterbarkeit	70
6.2.3	Autarkie.....	71
6.2.4	Verfügbarkeit.....	72
6.2.5	Grenzmaße/Grenzwert	72
6.2.6	Personen- und Geräteschutz	74
6.3	Allgemeiner Anwendungsfall	75
7	Fazit	78
7.1	Zusammenfassung	78
7.2	Ausblick	80
	Quellenverzeichnis	82

Abkürzungsverzeichnis

ADIRU	Air Data Inertial Reference Unit
ADM	Air Data Module
ARINC	Aeronautical Radio Incorporated
BSCU	Braking and Steering Unit
BTDM	Braking Temperature Data Module
DMC	Display Management Computer
DPU	Display Processing Unit
DU	Display Unit
ECP	EFIS/ECAM Control Panel
EFIS	Electronic Flight Instrument System
EIS	Electronic Instrument System
EWD	Engine/Warning Display
FCSB	Flight Computer System Bus
FWC	Flight Warning Computer
HW	Hardware
LGCIU	Landing Gear and Control Interface Unit
ND	Navigation Display
PFD	Primary Flight Display
ROS	Robot Operating System
SD	Status Display
SDAC	System Data Acquisition Concentrator
SW	Software
TPDM	Tire Pressure Data Module
TPIS	Tire Pressure Indicating System

Abbildungsverzeichnis

Abbildung 1: Schematische Darstellung von einem verteilten Computersystem. Vgl. [LaFe12]	4
Abbildung 2: Technisches System und Eingebettetes System. Vgl. [Siem12][Wenz18]	4
Abbildung 3: Eingebettetes verteiltes System im Zusammenhang. Vgl. [Siem12, Wenz18]	5
Abbildung 4: Verbindung von zwei ROS Knoten. Vgl. [Foot20]	6
Abbildung 5: Verbindung zwischen Kommunikationsteilnehmern bei ARINC. Vgl. [Cond00]	6
Abbildung 6: Bitzustände im zeitlichen Verlauf. Vgl. [Cond00]	7
Abbildung 7: Allgemeiner Aufbau eines ARINC 429 Datenwortes. [Cond00]	7
Abbildung 8: Verbindung zwischen Kommunikationsteilnehmern bei CAN. Vgl. [PaHe00]	8
Abbildung 9: CAN Paket aus der Sicht eines Programms. Vgl. [Meme00, Odin17, PaHe00]	9
Abbildung 10: Allgemeines Avionics Compartment. Vgl. [BsCa84, Flüh10]	9
Abbildung 11: ECAM und EFIS Systeme. [Ecam00, Ndim00, Pfdi00]	11
Abbildung 12: Zusammenhänge im EIS. Vgl. [Airb06]	12
Abbildung 13: Blockschaltbild des SDAC. Vgl. [Airb12]	12
Abbildung 14: Blockschaltbild des DMC. [Airb12]	13
Abbildung 15: Tire Pressure Indicating System. Vgl. [Airb96b]	14
Abbildung 16: Braking and Steering Presentation System. Vgl. [Airb96a]	15
Abbildung 17: ADIRU mit Sensoren. Vgl. [Airb96b]	15
Abbildung 18: Zusammenhang zwischen relevanten Systemen. Vgl. [Airb12, Airb96b, Airb96a] ..	16
Abbildung 19: Verbindungsstruktur der geforderten Flugsysteme	23
Abbildung 20: Übersicht über das grundlegende Strukturkonzept	31
Abbildung 21: CAD-Konstruktion des mechanischen Aufbaus	50
Abbildung 22: Schaltplan für die globale Stromversorgung	51
Abbildung 23: Die Umsetzung der globalen Stromverteilung	52
Abbildung 24: Schaltplan für die lokale Stromversorgung	52
Abbildung 25: Übersicht über die Systeme der Beispielanwendung	53
Abbildung 26: Visuelle Einheit des ADIRU Systems	53
Abbildung 27: Verbindungsstruktur der ADIRU Akquirierungseinheit	54
Abbildung 28: Aktivitätsdiagramm des ADM	54
Abbildung 29: Visuelle Einheit des BSCU/TPIS Systems	55
Abbildung 30: Verbindungsstruktur der BSCU und TPIS Akquirierungseinheit	55
Abbildung 31: Aktivitätsdiagramm des TPDM	56
Abbildung 32: Aktivitätsdiagramm des BTDM	56
Abbildung 33: Blockschaltbild des verteilten Systems	57
Abbildung 34: Belegung der Prozessebene durch die ADIRU Gruppe	58
Abbildung 35: Belegung der Prozessebene durch die Cockpit Gruppe (DMC, SDAC, DUs)	58
Abbildung 36: Belegung der Prozessebene durch das BSCU/TPIS Gruppe	58
Abbildung 37: Blockschaltbild der Treiberkette für den CAN-BUS Adapter	59
Abbildung 38: Reale ARINC 429 Softwareintegration	60
Abbildung 39: Verwendung von CAN für die Implementierung der Adressierungs- und Verbindungsfunktion	61
Abbildung 40: Bereitstellung der Verbindungstabelle für Avionics Computer	62
Abbildung 41: Aktivitätsdiagramm für die Initialisierung des Verbindungsmanagers	62
Abbildung 42: Blockschaltbild für die beispielhafte Anwendung des Kommunikationsmanagers	63
Abbildung 43: Beispielverbindungstabelle	63

Abbildung 44: Die Implementierung der geforderten Verbindungsstruktur	64
Abbildung 45: Verbindungstabelle des aktuellen Aufbaus.....	65
Abbildung 46: Brems temperaturtest.....	67
Abbildung 47: Latenzmessung mit Oszilloskop	68
Abbildung 48: Umgesetzter Aufbau des Demonstrators.....	69
Abbildung 49: Grafische Oberfläche des System Display	70
Abbildung 50: Messaufbau für die Gesamtstrommessung.....	71
Abbildung 51: Kontaktpunkte beim Anlehnen	75

Tabellenverzeichnis

Tabelle 1: Zusammenfassung der allgemeinen Anforderungen	26
Tabelle 2: Zusammenfassung der Anwendungsanforderungen	28
Tabelle 3: Zusammenfassung der Nachfolgeanforderungen	30
Tabelle 4: Vergleich von Abdeckungen.....	32
Tabelle 5: Vergleich von Materialansätzen für das Gerüst	33
Tabelle 6: Vergleich von Profilen für das Obergestell	33
Tabelle 7: Vergleich von Profilen für das Untergestell	34
Tabelle 8: Vergleich von Halterungssystemen	34
Tabelle 9: Vergleich von Middleware	35
Tabelle 10: Vergleich von ARINC 429 Ersatz	36
Tabelle 11: Vergleich der kombinierten Sensoren.....	37
Tabelle 12: Vergleich der Bremstemperatursensoren	37
Tabelle 13: Vergleich der Raddrucksensoren	38
Tabelle 14: Vergleich für den Ersatz eines Avionics Computers	39
Tabelle 15: Vergleich von Hutschienenadaptoren für die Avionics Computer	40
Tabelle 16: Vergleich der Betriebssysteme	40
Tabelle 17: Vergleich von Ersatzmöglichkeiten für Data Modules.....	41
Tabelle 18: Vergleich von lokalen Displays	42
Tabelle 19: Vergleich von Display Units.....	43
Tabelle 20: Vergleich von Stromversorgungssystemansätzen.....	43
Tabelle 21: Vergleich von Ansätzen für den Wechselstromanschluss	44
Tabelle 22: Vergleich von Hauptnetzteilen	44
Tabelle 23: Vergleich von lokalen Netzteilen	45
Tabelle 24: Vergleich von Verteilungsarten	46
Tabelle 25: Vergleich von Ansätzen für die lokale Spannungsverteilung	47
Tabelle 26: Vergleich des erwarteten Normalwerts und einer Momentaufnahme der Messwerte.....	66
Tabelle 27: Gesamtstrom bei drei Betriebsfällen	72
Tabelle 28: Vergleich der Maße der Präsentationsebene.....	73
Tabelle 29: Vergleich der Gesamtmaße des Demonstrators	73
Tabelle 30: Schätzung des Gewichtes der tragenden und abdeckenden Mechanik.....	74

1 Einleitung

In diesem Kapitel wird in das Thema eingeleitet und das Ziel der Arbeit definiert. Es wird auch der Rahmen und die Vorgehensweise beschrieben.

1.1 Motivation

In der Welt der Luftfahrt, wie in jedem anderen fortschreitenden technischen Bereich, werden neuartige elektronische Systeme entwickelt, um die bestehenden Umsetzungen zu optimieren und zu ersetzen. Um die vorgesehenen Funktionen zu verifizieren und zu validieren, durchläuft das neue System bisher einen langwierigen und kostspieligen Prozess, bevor es erstmalig in einem Flugzeug in die bestehenden Systeme integriert werden kann. Da sich aber die moderne Welt aus technischer Sicht rasant ändert und kontinuierlich neue Technologien auf den Markt kommen, ist dieser methodische Ansatz nicht mehr dafür geeignet. Neue Ideen sollen schnell und effizient bewertet werden können. Es wird ein technologischer Ansatz benötigt, der die Grundfunktionen der elektronischen Flugsysteme nachbildet, aber flexibel genug ist, um neue Ideen bewerten bzw. verifizieren zu können.

1.2 Ziel

Das Ziel dieser Bachelorarbeit ist es, eine sogenannte Flugzeugreferenzarchitektur in Form eines elektro-mechanischen Demonstrationsstandes aufzubauen. Dabei soll das *Avionics Compartment*¹ eines Airbus A320 Passagierflugzeuges funktionell nachgebildet werden. Die Nachbildung dient als Grundlage für die Forschung an neuartigen Wartungssystemen.

Der Aufbau soll Wissenschaftlern im Bereich der Luftfahrtinstandhaltung die Möglichkeit bieten, Forschungsthemen modellbasiert und mit vermindertem Aufwand und geringeren Kosten entwickeln, verifizieren und implementieren zu können. Darüber hinaus soll er den Forschern ermöglichen, ihre Ergebnisse anschaulich und greifbar zu präsentieren.

Die Flugzeugreferenzarchitektur bildet eine Grundlage für zukünftige Projekte. So soll im Kontext dieser Arbeit vor allem die Durchgängigkeit² einiger Flugdaten demonstriert werden. Dazu soll neben dem eigentlichen Demonstrationsstand eine Beispielanwendung mit insgesamt vier Mess- und Anzeigesystemen modular nachgebildet werden: *Electronic Instrument System (EIS)*³, *Air Data Inertial Reference Unit (ADIRU)*⁴, *Tire Pressure Indicating System (TPIS)*⁵ und *Braking and Steering Control Unit (BSCU)*⁶. Diese Systeme sollen Daten von Sensoren sammeln sie über einen partiell nachgebildeten *ARINC 429 Avionikbus*⁷ versenden. Im Anschluss sollen diese Daten auf geeigneten Displays angezeigt werden.

¹ Zentraler Ort für die Elektronik eines Verkehrsflugzeuges. (Siehe 2.3)

² Vorschau eines Messsystems.

³ Das Anzeige- und Aufnahmesystem von dem Avionics Compartment. (Siehe 2.4)

⁴ Luftdaten- und Trägheitsnavigationssystem. (Siehe 2.5.3)

⁵ Messsystem für den Raddruck. (Siehe 2.5.1)

⁶ Messsystem für die Bremsstemperatur. (Siehe 2.5.2)

⁷ Kommunikationsbussystem in einem Flugzeug. (Siehe 2.2.1)

1.3 Rahmen

Die Bachelorarbeit wird im Rahmen eines Projektes am Institut für Instandhaltung und Modifikation, einem Institut des Deutschen Zentrums für Luft und Raumfahrt e.V., geschrieben. Die funktionellen Vorgaben für den Demonstrator bzw. Demonstrationsstand werden von den Projektanforderungen abgeleitet. Die Ausarbeitung wurde durch einen engen Austausch mit Experten auf dem Gebiet der Luftfahrttechnik begleitet.

1.4 Vorgehensweise

In Kapitel 1 wird das Thema der Arbeit vorgestellt und die Zielsetzung definiert.

In Kapitel 2 werden die für diese Arbeit relevanten Technologien beschrieben. Dabei handelt es sich um technische Konzepte, die hinter der Avionik stehen. Dementsprechend werden die wichtigsten Avionik Systeme vorgestellt. Für die Recherche wird die luftfahrtrelevante Literatur sowie einige Vorlesungsskripte für verteilte und eingebettete Systeme untersucht.

In Kapitel 3 werden die Anforderungen an den Demonstrator beschrieben. Es werden Interviews mit den Projektbeteiligten geführt, um die Bedarfe festzustellen. Auf Grundlage der Interviews werden sinnvolle Annahmen getroffen, um ein technisches System realisieren zu können. Hierbei wird eine Beispielanwendung definiert die es erlaubt, die Funktion des Demonstrators verifizieren zu können.

In Kapitel 4 wird das Strukturkonzept des Demonstrators konkretisiert. Es werden Vergleiche auf der Grundlage der technischen Spezifikation durchgeführt und geeignete Technologien für den Demonstrator ausgewählt.

Kapitel 5 beschreibt die Umsetzung des Demonstrators und der Beispielanwendung. Es werden Schaltpläne entworfen und mit den in Kapitel 4 gewählten Komponenten umgesetzt.

In Kapitel 6 wird der Demonstrator auf die Erfüllung der Spezifikation und des Ziels verifiziert. Hierbei wird die Beispielanwendung separat und die Demonstratortechnik insgesamt untersucht. Schließlich wird ein allgemeiner Anwendungsfall definiert um zu prüfen, ob das Ziel dieser Arbeit erfüllt wurde.

Kapitel 7 beinhaltet eine Zusammenfassung der Arbeit und liefert einen Ausblick für die zukünftige Entwicklung des Demonstrators.

2 Ausgangssituation

In diesem Teil der Arbeit werden Technologien beschrieben, die für den Aufbau der Flugzeugreferenzarchitektur⁸ benötigt werden. Das sind die Grundlagen der verteilten und eingebetteten Systeme (Abschn. 2.1), die Computer- und Kommunikationstechnik relevanter Flugzeugsysteme⁹ (Abschn. 2.2 - 2.5) und der Zusammenhang zwischen diesen Systemen (Abschn. 2.6).

2.1 Grundlagen

In diesem Unterkapitel werden die Grundlagen beschrieben, auf denen das Avionics Compartment aufbaut. Zusätzlich wird das Thema *Robot Operating System (ROS)* erwähnt, das für den Aufbau des Demonstrators benötigt wird.

2.1.1 Verteilte Systeme

Nach [Bapa94] und [DiHa03] sind verteilte Systeme folgenderweise definiert: „Ein verteiltes System ist ein informationsverarbeitendes System, das eine Vielzahl von eigenständigen Rechnern enthält, die über ein Kommunikationsnetzwerk miteinander kooperieren, um ein angestrebtes Ziel zu erreichen.“

Nebenläufigkeit und getrennter Arbeitsspeicher

Die Softwarekomponenten eines verteilten Systems sind natürlicherweise nebenläufig, da jede Komponente einen eigenen Arbeitsspeicher und einen eigenen Prozessor besitzt. Die Komponenten haben hierbei gemeinsame Ressourcen wie Daten und Zustände, auf die sie zugreifen und die sie in ihren Prozessen verwenden. Vgl. [Ngai09]

Kommunikation über Nachrichtenpakete und Prozeduraufrufe

Die Komponenten arbeiten in getrennten Arbeitsspeichersegmenten. Um dennoch Nutzinformationen austauschen zu können, werden Nachrichten gesendet oder entfernte Prozeduraufrufe ausgeführt. Die Nachrichten oder Prozeduraufrufe werden über ein Kommunikationsmedium und -protokoll an die Zielkomponenten gesendet. Im Falle eines Prozeduraufrufs warten zusätzlich die Zielkomponenten auf Ergebnisse. Vgl. [LaFe12][Ngai09]

Middleware für die Abstrahierung des Systems

Netzwerke, Computer-Hardware, Betriebssysteme und Programmiersprachen können sich bei verteilten Systemen grundlegend unterscheiden. Die Unterschiede dieser Komponenten werden durch eine *Middleware* maskiert und abstrahiert. Sie ist dafür zuständig, den Zugriff auf das verteilte System für Benutzer und Anwendungen transparent zu machen. Die Middleware wird zwischen der Benutzeranwendung und dem Betriebssystem implementiert. Die Middleware hat auf unterschiedlichen Computern die gleiche Schnittstelle für die Benutzeranwendung, wodurch die Anwendung keine Informationen über die Details der Verteilung besitzen muss. Vgl. [Ngai09][LaFe12][John15]

⁸ Flugzeugreferenzarchitektur - Modell der Avionik eines Verkehrsflugzeugs.

⁹ Siehe 2.3

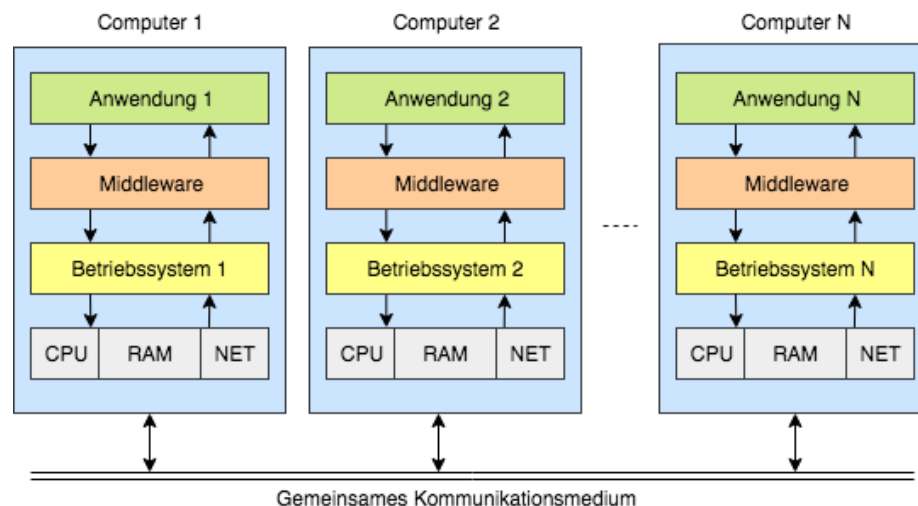


Abbildung 1: Schematische Darstellung von einem verteilten Computersystem. Vgl. [LaFe12]

In Abbildung 1 ist der schematische Aufbau von Computern in einem verteilten System dargestellt. Alle Computer sind miteinander über ein Kommunikationsmedium verbunden. Die Hardware der Computer (CPU, RAM, NET) kommuniziert mit dem Betriebssystem. Das Betriebssystem kommuniziert mit der Middleware. Die Middleware kommuniziert mit der Anwendung. Über die Middleware werden Nachrichtepakete zwischen den Softwareanwendungen der Computer im verteilten System ausgetauscht. Vgl. [John15][LaFe12]

2.1.2 Eingebettete Systeme

Nach [Siem12] sind eingebettete Systeme folgenderweise definiert: „Eingebettete Systeme (*embedded systems*) sind Computersysteme, die aus HW und SW bestehen und die in komplexe technische Umgebungen eingebettet sind. Diese Umgebungen sind meist maschinelle Systeme, in denen das eingebettete System mit Interaktion durch einen Benutzer arbeitet oder auch vollautomatisch (autonom) agiert. Die eingebetteten Systeme übernehmen komplexe Steuerungs-, Regelungs- und Datenverarbeitungsaufgaben für bzw. in diesen technischen Systemen.“

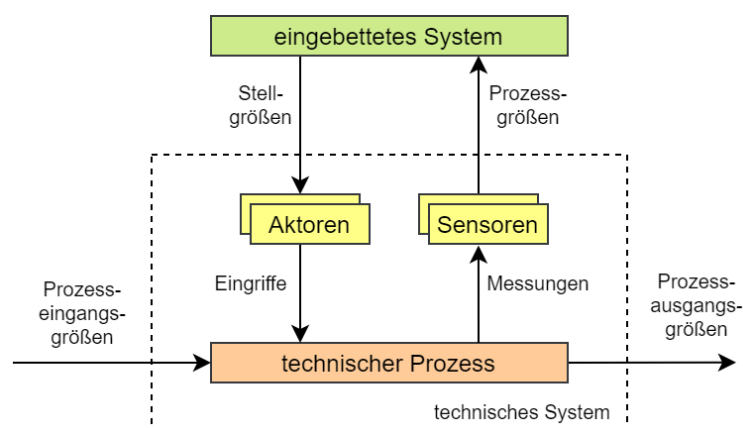


Abbildung 2: Technisches System und Eingebettetes System. Vgl. [Siem12][Wenz18]

Abbildung 2 stellt schematisch die Wechselwirkung zwischen einem technischen und einem eingebetteten System dar. Ein eingebettetes System erzeugt Stellgrößen, die über die Aktoren auf den technischen Prozess einwirken. Prozessgrößen werden über Sensoren gemessen und an das eingebettete System gesendet. Der in Abbildung 2 dargestellte technische Prozess nimmt

Prozesseingangsgrößen wie Materie, Energie und vergleichbare Informationen auf. Dieser verarbeitet sie und gibt Prozessausgangsgrößen wieder aus. Akteure, Sensoren und der technische Prozess machen das technische System aus. Vgl. [Siem12, Wenz18]

Ein verteiltes eingebettetes System stellt ein System von gekoppelten Rechnern dar. Die Rechner sind räumlich eng beieinander und werden in ein übergeordnetes technisches System integriert. Die Abbildung 3 zeigt ein verallgemeinertes verteiltes System, das in ein technisches System eingebettet wurde. Einzelne eingebettete Systeme kommunizieren miteinander über ein Kommunikationsmedium und wirken auf ihren Teil des technischen Systems. Vgl. [Siem12]

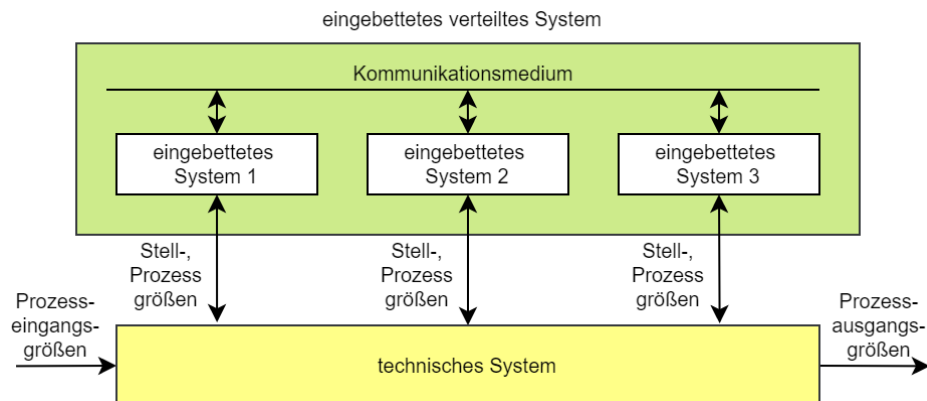


Abbildung 3: Eingebettetes verteiltes System im Zusammenhang. Vgl. [Siem12, Wenz18]

2.1.3 Robot Operating System

Das *Robot Operating System (ROS)* ist ein quelloffenes software-technisches Ökosystem, das unter anderem für die Entwicklung neuer Roboteranwendungen verwendet wird. Sie basiert auf dem Konzept eines *Rechengraphs*¹⁰. Die Knoten sind Programme, die eine „Main“ besitzen. Die Knoten kommunizieren miteinander über Kanten. Die Kommunikation über Kanten basiert auf dem Konzept „Publisher-Subscriber“ Programmiermuster. Vgl. [Foot00][Coll19]

Grundlegende Kommunikationsinfrastruktur

Aus der Sicht der verteilten Systeme ist die grundlegende Kommunikationsinfrastruktur von ROS eine verteilte Middleware. ROS kann verwendet werden, um die Kommunikation zwischen den Programmen und ihren Rechnern zu abstrahieren. Beim Design einer Anwendung kann sich der Entwickler verstärkt auf die aufgabenspezifische Lösung konzentrieren. Die niederen Ebenen der Kommunikation werden dabei vor dem Benutzer verborgen. Vgl. [Ros00]

Der *Nachrichtenaustausch* basiert auf dem Konzept „*Publisher-Subscriber*“. Die sogenannten *Publisher* senden asynchron Nachrichten an *Topics*. *Subscriber* empfangen Nachrichten von den *Topics*. Ein Programm kann sowohl ein *Publisher* als auch ein *Subscriber* sein. Beide können senden und empfangen ohne voneinander Kenntnis zu haben. Für die Verbindung der einzelnen Knoten ist ein *ROS Master* zuständig. Abbildung 4 zeigt exemplarisch die Verbindung zwischen zwei Knoten über ein *Topic* mit Hilfe eines *ROS Masters*. (vgl. [Foot00])

¹⁰ Bei klassischen Rechengraphen sind Knoten mathematische Operationen oder Variablen. Die Daten werden über Kanten zwischen den Knoten weitergegeben. [Dani17]

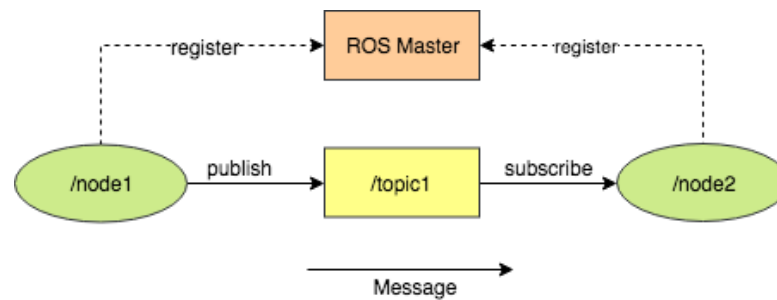


Abbildung 4: Verbindung von zwei ROS Knoten. Vgl. [Foot20]

Werkzeuge

Zu den Werkzeugen gehören nach [Ros00] drei Systeme:

- Kommandozeilenwerkzeuge.
- 3D visualization tool – *rviz*.
- Qt-based Framework für GUI – *rqt*.

Mit den *Kommandozeilenwerkzeugen* wird eine ROS Sitzung gesteuert und analysiert. Jedes einzelne Programm kann gestartet, inspiziert und gestoppt werden. Jedes Topic und jede Nachricht können auf mögliche Fehler analysiert werden. *rviz* ist für die Visualisierung der Raumdaten eines Roboters zuständig. *rqt* kann interne Zustände darstellen und auf sie einwirken. *rviz* und *rqt* lassen sich über ein standardisiertes *Pluginsystem* leicht erweitern. Vgl. [Ros00]

2.2 Kommunikationsbusse in der Avionik

In der Welt der Avionik gibt es eine große Anzahl von Kommunikationsbussen. Jeder Bus ist für einen bestimmten Anwendungszweck entwickelt worden. In Rahmen dieser Arbeit werden zwei verbreitete Avionikbusse verwendet, nämlich der ARINC 429 und der CAN-Bus.

2.2.1 ARINC 429

ARINC 429 ist ein weit verbreiteter und höchst zuverlässiger Datenbus bei Verkehrsflugzeugen. Er wurde nach dem *Mark 33 Digital Information Transfer System (DITS)* Standard definiert. *ARINC 429* versendet Daten seriell und unidirektional (1-n). Ein Sendeausgang kann mehrere Empfängereingänge versorgen. Jede Verbindung besteht aus verdrehten Drahtpaaren, die differentielle Signale übertragen. Für das gleichzeitige Empfangen und Senden zwischen zwei Teilnehmern werden zwei Drahtpaare benötigt. Vgl. [Cond00]

Abbildung 5 zeigt einen Beispielaufbau zwischen drei Kommunikationsteilnehmern. Teilnehmer 1 verschickt Daten an Teilnehmer 2 und Teilnehmer 3 über denselben Ausgangskanal. Lediglich Teilnehmer 2 sendet Daten zurück über einen separaten Rückkanal.

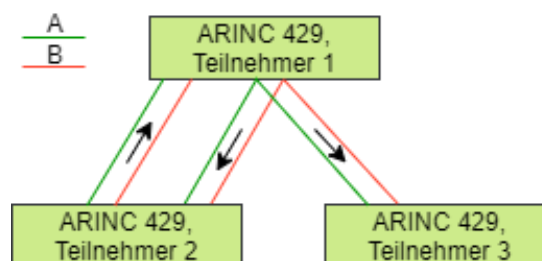


Abbildung 5: Verbindung zwischen Kommunikationsteilnehmern bei ARINC. Vgl. [Cond00]

Das ARINC 429 Busprotokoll verwendet „Bipolar Return-to-Zero“ Linienkodierung. Ein Informationsbit (HI oder LO) wird durch eine positive (10 V) oder negative (-10 V) Nominalspannung repräsentiert. Die Taktung erfolgt mithilfe eines Nullzustands (0 V), der zwischen den Bits eingefügt wird. Diese Art der Synchronisation nennt man „self-clocking“¹¹. In Abbildung 6 sieht man den zeitlichen Verlauf der Bitreihenfolge „1-0-1“. Die Nominalspannungen sinken, wenn mehrere Empfänger an einen Ausgang angeschlossen werden. Ein Ausgang kann maximal 20 Empfänger versorgen. Vgl. [Cond00]

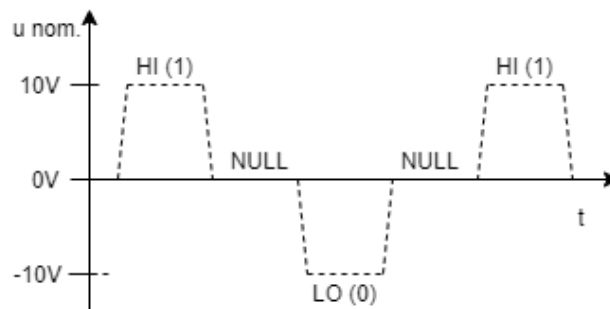


Abbildung 6: Bitzustände im zeitlichen Verlauf. Vgl. [Cond00]

Die Daten können entweder mit einer Datenrate von 12,5 Kbps oder 100 Kbps gesendet werden. Jedes gesendete Datenwort hat eine Länge von 32 Bits. Zwischen zwei aufeinander folgenden Worten gibt es eine Pause von 4 Bits. Eine Pause wird durch den Nullzustand kodiert. Vgl. [Cond00]

Ein Datenwort nach [Cond00] besteht aus den fünf Feldern:

- Parity (P),
- Sign/Status Matrix (SSM),
- Data,
- Source Destination Identifier (SDI) und
- Label.

Abbildung 7 stellt ein allgemeines ARINC 429 Datenwort dar. Für „Parity“ wurde das Bit 32 reserviert. Für „SSM“ wurden die Bits 31 bis 30 reserviert. Für „Data“ die Bits 29 bis 11. Für „SDI“ die Bits 10 bis 9. Für „Label“ die Bits 8 bis 1. Der Benutzer des ARINC 429 Busses muss alle fünf Felder selbst füllen. Vgl. [Cond00]

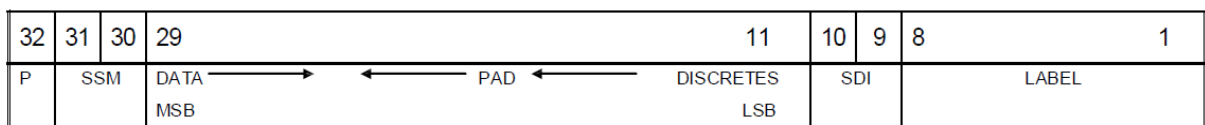


Abbildung 7: Allgemeiner Aufbau eines ARINC 429 Datenwortes. [Cond00]

Die ARINC Datenwortfelder haben nach [Cond00] die folgende Bedeutung:

- „Parity“ wird für die Validierung der Übertragung eines Datenwortes verwendet.
- „SSM“ repräsentiert mehrere Informationen: Das Vorzeichen einer Zahl, den Geräte- und Betriebszustand.

¹¹ Die Verbindung benötigt keine zusätzliche Taktleitung.

- „Data“ trägt Nutzinformationen in Form von Zahlen oder diskreten Zuständen.
- „SDI“ wird für die Unterscheidung der Paketziele verwendet.
- „Label“ wird für die Unterscheidung einzelner Nachrichten verwendet. „Label“ trägt zusätzlich bei dem BCD Datenformat die Position des Kommas.

2.2.2 CAN-Bus

Das *Controller Area Network* (CAN) ist ein Datenbus, das für steuerungstechnische Anwendungen entwickelt wurde. Der CAN-Bus hat seinen Ursprung in der Automobilindustrie und kann in einer induktiv belasteten Umgebung eingesetzt werden. Dadurch kann man mit CAN sicher und zuverlässig Daten zwischen Sensoren bzw. Aktuatoren und den jeweiligen Steuerungen austauschen. Vgl. [Odin17, PaHe00]

Das CAN ist ein *Multimaster* und *Linientopologie*-Bus. Das bedeutet, dass jeder Teilnehmer zu einem Master werden und an alle anderen Teilnehmer Daten senden kann. Wenn mehrere Teilnehmer gleichzeitig senden wollen, entscheidet ein spezieller *Arbitrierungsmechanismus* (CSMA/CR Verfahren) die Reihenfolge des Versands. Die Adressierung findet über *IDs* statt. Die Empfänger filtern auf Basis der *ID* die Nachrichtenpakete. Durch die Filterung können die Teilnehmer selbst entscheiden, von welchen IDs sie Daten empfangen. Vgl. [Meme00, Odin17, PaHe00]

In Abbildung 8 sind drei Kommunikationsteilnehmer miteinander verbunden. Jeder Teilnehmer besitzt einen Rechner, einen CAN Controller und einen CAN Treiber. Der Rechner stellt die zu sendende Information zur Verfügung. Der CAN Controller bereitet ein CAN Paket aus den Nutzinformationen vor. Der CAN Treiber liefert die nötigen differenziellen Spannungen für das CAN Paket. Jeder Teilnehmer wird durch die CAN Treiber über ein gemeinsames verdrehtes Drahtpaar (CAN HIGH und CAN LOW) verbunden. Das verdrehte Drahtpaar überträgt ein differenzielles Signal. Jeder CAN Treiber muss mit einem 120 Ohm Widerstand terminiert werden. Durch entsprechende CAN Treiber können bis zu 128 Busteilnehmer in einer Netzebene verbunden werden. Vgl. [Meme00, Odin17, PaHe00]

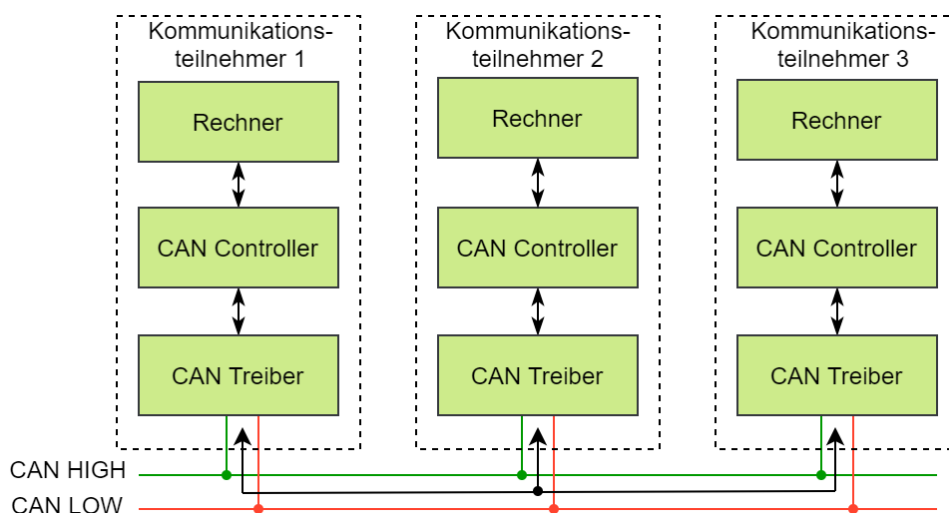


Abbildung 8: Verbindung zwischen Kommunikationsteilnehmern bei CAN. Vgl. [PaHe00]

Bei der Benutzung des CAN-Busses werden nur drei Felder gefüllt. Dabei handelt es sich um die Felder *Identification* (ID), *Data-Length-Code* (DLC), *DATA*. Die Abbildung 9 stellt ein CAN Datenpaket

aus der Sicht eines Programms dar. Die ID dient der Identifizierung des Paketes. Dieses Feld kann die Länge von 11 Bits oder 29 Bits haben. Anhand der ID filtern die Empfänger die Nachrichten. Das Feld DLC stellt die Größe der zu übertragenden Daten dar und hat eine Größe von 4 Bytes. Das Feld DATA stellt die Nutzinformationen dar. DATA kann in der Größe variieren aber maximal 8 Bytes groß sein. CAN Geräte können im Standardformat 11 Bits für die ID verwenden und im erweitertem Format 29 Bits. Für die Benutzung des erweiterten Formates wird vom Programm noch zusätzlich das Extended ID Bit gesetzt. Weitere Felder wie CRC, Start, End werden von dem CAN Controller verwaltet. Vgl. [Meme00, Odin17, PaHe00]

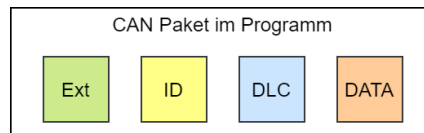


Abbildung 9: CAN Paket aus der Sicht eines Programms. Vgl. [Meme00, Odin17, PaHe00]

2.3 Avionics Compartment

Das *Avionics Compartment* ist ein zentraler Bestandteil der Elektronik eines Flugzeuges. Dabei handelt es sich um einen Raum unterhalb des Cockpits. Dieser beinhaltet mehrere Schränke für Computersysteme und Verbindungskabel. Die Computer können durch ein normiertes Einschubsystem ausgetauscht werden. Durch den zentralen Ort der Rechner können Tests und Wartungsarbeiten mit guter Zugänglichkeit durchgeführt werden. Vgl. [Airb12]

In Abbildung 10 sieht man den allgemeinen Aufbau eines Avionics Compartment. Die Flugsensordaten werden gesammelt und an das Computersystem des Avionics Compartments übergeben. Die Computer verarbeiten diese Daten und erstellen Zustandswerte eines Flugzeuges für die Anzeigen des Cockpits. Der Pilot nimmt die aktuellen Flugzeugzustände wahr und erzeugt bei Erforderlichkeit seinerseits Steuereingriffe. Diese werden von dem Avionics Compartment verarbeitet und entsprechende Stellbefehle für die Aktuatoren generiert. Die Sicherheitsmechanismen oder der Autopilot können Stellbefehle für die Aktuatoren auch direkt erzeugen. Die Aktuatoren greifen in den physikalischen Verlauf des Fluges ein. Vgl. [Airb12]

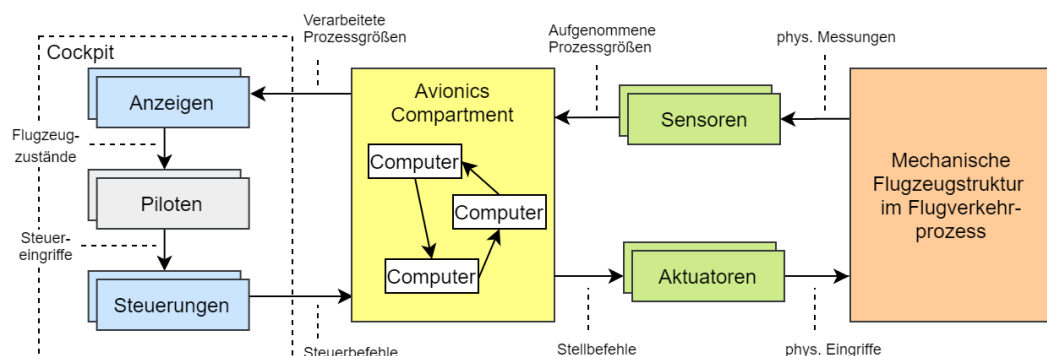


Abbildung 10: Allgemeines Avionics Compartment. Vgl. [BsCa84, Flüh10]

Das Computersystem des Avionics Compartments beruht auf dem Konzept der verteilten eingebetteten Systeme. Jeder Computer hat eine spezifische Aufgabe. Sie kommunizieren über einen Avionik-Bus und tauschen Daten aus. Sie alle haben das Ziel den Flugverlauf korrekt zu führen. Einige Computersysteme sind mehrfach vorhanden und kommunikationstechnisch

parallelgeschaltet. Dadurch kann der Pilot beim Ausfall der Hauptsysteme auf die funktionstüchtigen redundanten Systeme zugreifen. Vgl. [Airb12, Flüh10]

Einige Computer basieren auf der *Motorola 68k CPU-Serie*. Ein eingesetzter Vertreter ist der MC68020. Es ist ein 32 Bits extern adressierbarer CISC¹² Mikroprozessor. Es ist möglich, einen bis zu 4GB großen Arbeitsspeicher zu integrieren. Weiterhin benötigt diese CPU eine externe Peripherie, um mit externen Geräten kommunizieren zu können. Die Taktfrequenzen liegen im Bereich von 12,5 MHz bis 33,33 MHz. Vgl. [Cpu20]

2.4 Electronic Instrument System

Das *Electronic Instrument System* (EIS) ist ein System für das Zusammenführen, Verarbeiten und Anzeigen von sämtlichen relevanten Flugdaten und Zuständen des Flugzeuges. Es bildet die Grundlage für die Informationsbereitstellung für den Piloten.

Die Funktionen des EIS werden durch eine Reihe von speziellen Systemen verrichtet. Diese Systeme sind untereinander verbunden und tauschen Daten über einen Avionik-Bus oder diskrete Leitungen aus. Die kritischen Systeme sind redundant ausgelegt und können bei Bedarf im Betrieb umgeschaltet werden. Vgl. [Aust08]

Es gibt nach [Aust08] vier grundlegende elektronische Gruppen beim EIS:

- Die Avionics Computer; zu den Avionics Computer gehören:
 - *Der Display Management Computer (DMC)*
 - *Der System Data Acquisition Concentrator (SDAC)*
 - *Der Flight Warning Computer (FWC)*
- Die Control Panels (ECP); dazu gehören:
 - *Das EFIS Control Panel*
 - *Das ECAM Control Panel.*
- Die Display Units (DU), zu den Display Units gehören:
 - *Das Primary Flight Display (PFD)*
 - *Das Navigation Display (ND)*
 - *Das Engine/Warning Display (EWD)*
 - *Das System Display (SD)*
- Die Attention Getters (AG), dazu gehören
 - *Lautsprecher*
 - *Warnleuchten*

Avionics Computer führen Berechnungen aus. Diese Computer befinden sich im Avionics Compartment. Control Panels dienen der Konfiguration der EFIS und ECAM Systeme, diese sind unmittelbar vor den Händen der Piloten. Display Units zeigen die Flug- und Flugzeuginformationen an und bestehen aus 7.25" x 7.25" große Kathodenstrahlröhren. Die Attention Getter lösen aus, falls eine kritische Situation eintritt. Das Auslösen findet durch Lautsprecher und speziell beschriftete Lampen statt. [Aust08][Airb12]

Das EIS ist in zwei Subsysteme aufgeteilt [Aust08]:

- *Electronic Flight Instrument System (EFIS).*
- *Electronic Centralized Aircraft Monitor (ECAM).*

¹² CISC - Complex Instruction Set Computing.

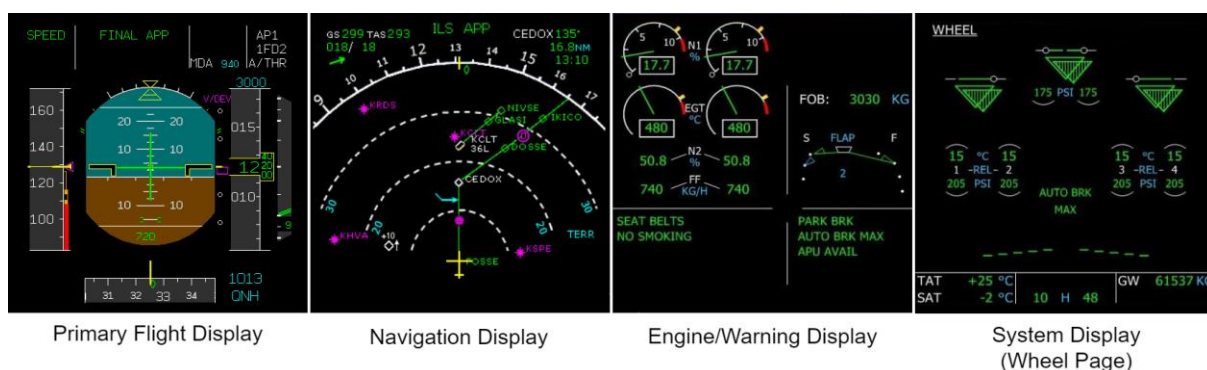


Abbildung 11: ECAM und EFIS Systeme. [Ecam00, Ndim00, Pfdi00]

Abbildung 11 zeigt einige Ausschnitte der Cockpitdisplays. Das Primary Flight Display (PFD)¹³ und das Navigation Display (ND)¹⁴ bilden zusammen das EFIS. Das Engine/Warning Display (EWD)¹⁵ und das System Display (SD)¹⁶ bilden das ECAM System. Das EFIS hat die Aufgabe, Navigations- und Wetterdaten bereitzustellen. Der Pilot kann mithilfe dieser Daten einen Flug von A nach B berechnen lassen und leiten. Es werden hierbei Lage, Luftgeschwindigkeit und Flughöhe angezeigt sowie weiteren Parameter. ECAM hat die Funktion, übersichtlich und geordnet jegliche Funktionszustände des Flugzeuges anzuzeigen. Diese werden im System Display auf den sogenannten Pages angezeigt. Zum Beispiel zeigt das Wheel Page Raddruck, Bremsen- und den Zustand der ein- oder ausgefahrenen Räder an. Es werden auf allen Pages statische Temperatur und Gesamtlufttemperatur angezeigt. [Aust08][Airb12]

Abbildung 12 stellt den Zusammenhang zwischen einzelnen Systemen im EIS dar. Die Sensor- und Zustandswerte des Flugzeuges werden von dem DMC, dem FWC und dem SDAC aufgenommen und verarbeitet. In Warnsituationen werden Daten vom SDAC an den FWC gesendet, damit der FWC Warnmeldungen für den DMC generieren kann. Einige Daten des SDAC werden an den DMC direkt gesendet. Der DMC bereitet Grafiken aus allen empfangenen Daten für die Display Units vor. Die Daten können dann auf den Display Units der Piloten, dem PFD, ND, EWD und SD angezeigt werden. Attention Getter werden ausgelöst, wenn bestimmte Grenzwerte überschritten werden. Vor allem bei nicht erlaubten Konfigurationen des Flugzeuges. Über die ECPs können Einstellungen an den ECAM und EFIS Systemen vorgenommen werden. Die ECAM und EFIS Systeme sind redundant ausgeführt und können im Betrieb getauscht werden. Die Systeme des Kapitäns und des ersten Offiziers sind voneinander unabhängig. Bei Ausfall eines Systems bleibt das andere intakt. Vgl. [Airb06][Airb12]

¹³ Primary Flight Display – Zeigt Informationen über Fluglage, Fluggeschwindigkeit und Flughöhe. [Airb12]

¹⁴ Navigation Display – Zeigt Informationen über die Flugroute, andere Flugzeuge, Wind und Richtung. [Airb12]

¹⁵ Engine/Warning Display – Zeigt Warnungen an und informiert über den Zustand der Triebwerke. [Airb12]

¹⁶ System Display – Zeigt Informationen von unterschiedlichen Flugsystemen auf verschiedenen speziellen Seiten. [Airb12]

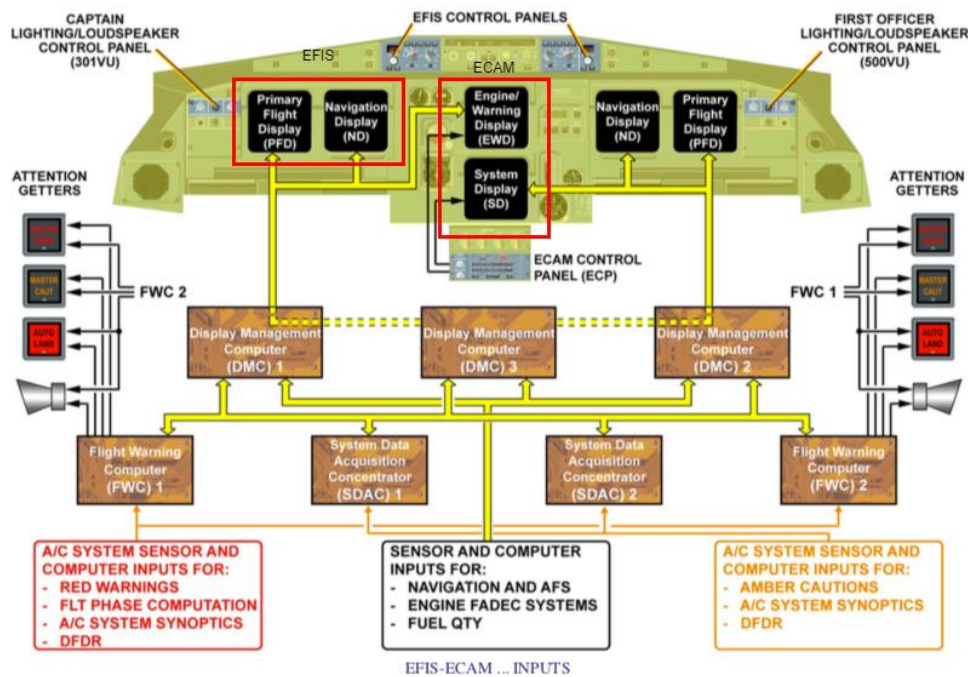


Abbildung 12: Zusammenhänge im EIS. Vgl. [Airb06]

2.4.1 System Data Acquisition Computer

Der *System Data Acquisition Concentrator* (SDAC) ist dem EIS zugeordnet. Bei bestimmten Warn- und Alarmsituationen werden Fehlfunktionsdaten von diversen Flugsystemen generiert. Diese Daten werden von dem SDAC akquiriert, digitalisiert und konzentriert. Die digitalisierten und konzentrierten Daten werden an den Flight Warning Computer (FWC) und den Display Management Computer (DMC) gesendet. Der FWC generiert daraus Alarm- und Vorgehensmeldungen als Hilfestellung für den Piloten. Der DMC generiert daraus animierte synoptische Diagramme von den Systeminformationen und Triebwerksparametern. Vgl. [Airb12]

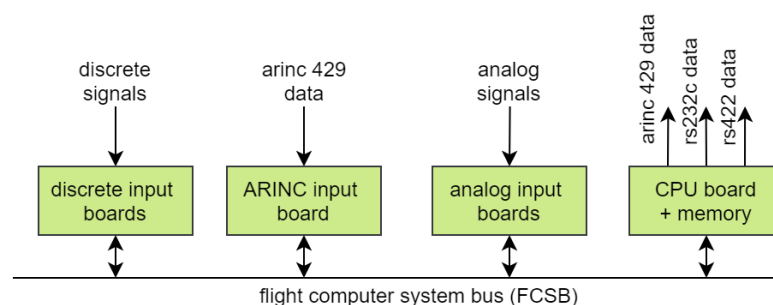


Abbildung 13: Blockschaftbild des SDAC. Vgl. [Airb12]

Abbildung 13 zeigt ein vereinfachtes Blockschaftbild der internen Funktion eines SDAC. Die Daten werden auf drei grundlegende Arten von dem jeweiligen Eingangsmodul akquiriert: Die diskreten Signale durch das diskrete Modul, die analogen Signale durch das analoge Modul und die Bus-Daten durch das ARINC 429 Modul. Die analogen Signale werden zusätzlich konditioniert und digitalisiert. Das CPU-Board koordiniert diesen Vorgang und tauscht Daten mit dem jeweiligen Modul über den Flight Computer System Bus (FCSB) aus. Die Daten werden anschließend über das CPU-Board an den DMC oder FWC gesendet. Dabei stehen drei Kommunikationsarten zu Verfügung: ARINC 429, RS232C und RS422. Vgl. [Airb12]

2.4.2 Display Management Computer

Der *Display Management Computer* (DMC) ist dem EIS zugeordnet. Seine Aufgabe besteht darin, Daten von diversen Flugzeugsystemen zu konzentrieren und für die DUs vorzubereiten. Vgl. [Airb06][Airb12]

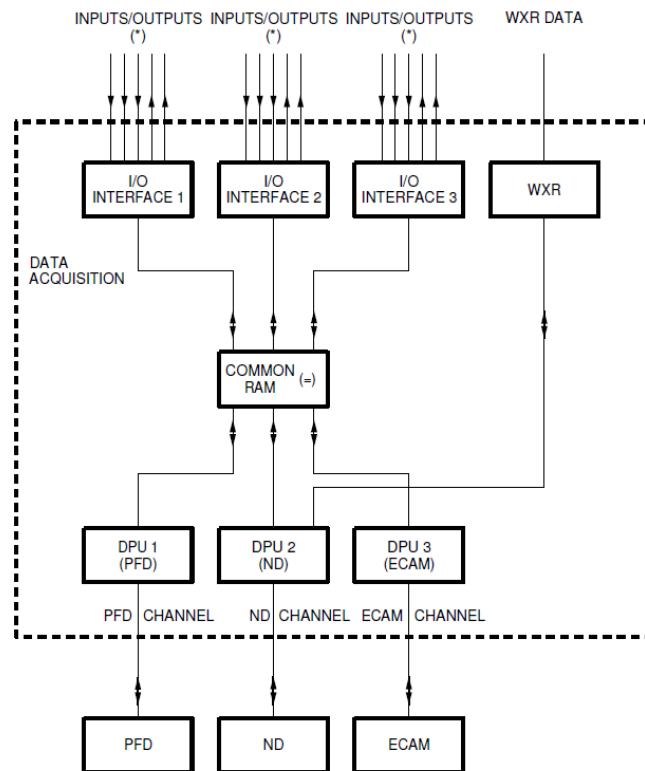


Abbildung 14: Blockschaltbild des DMC. [Airb12]

Die Abbildung 14 zeigt ein Blockschaltbild eines DMC. Die gestrichelte Linie stellt die Systemgrenzen des DMC dar. Innerhalb der gestrichelten Linie kann man die einzelnen Bestandteile und ihre Verbindungsstruktur sehen. Der DMC besitzt die folgenden grundlegenden Komponenten: Input/Output Interfaces (I/O Interface), WXR Interface, Display Processing Units (DPU) und gemeinsamer Arbeitsspeicher (COMMON RAM). Die I/O und WXR Interfaces können folgende Verbindungsprotokolle aufnehmen: Diskret, ARINC 429, ARINC 453, RS232, RS422. Vgl. [Airb06][Airb12]

Jede DPU ist ein separates Board mit einem Mikroprozessor und eigenem Arbeitsspeicher. Eine DPU ist jeweils zuständig für ihre Display Unit. Insgesamt gibt es nach [Airb12] drei DPUs:

- PFD DPU - zuständig für die PFD Anzeige,
- ND DPU – zuständig für ND Anzeige und
- ECAM DPU – zuständig für einen der zwei ECAM Anzeigen.

Der *gemeinsame Arbeitsspeicher* (COMMON RAM) befindet sich in der PFD DPU. Vgl. [Airb06][Airb12]

Durch die Interfaces werden Daten und Signale von externen Systemen empfangen. Die Signale sind Mess- und Zustandsgrößen des Flugzeuges. Die empfangenen Daten werden in dem gemeinsamen Arbeitsspeicher (COMMON RAM) abgelegt. DPUs akquirieren die Daten aus dem

gemeinsamen Arbeitsspeicher und generieren Grafikbefehle und Codes für die DUs. Diese werden an die DUs über den ARINC 429 Bus gesendet. Die DUs bereiten synoptische¹⁷ Diagramme vor und stellen sie den Piloten zu Verfügung. Vgl. [Airb06][Airb12]

2.5 Weitere Flugzeugsysteme

In diesem Unterkapitel werden Flugsysteme beschrieben, die nicht Teil der Electronic Instrument Systems sind.

2.5.1 Tire Pressure Indicating System

Das *Tire Pressure Indicating System* (TPIS) ist ein Teil des Landing Gear Systems. Das TPIS nimmt Druckinformation für jedes Rad auf und sendet es an das ECAM System. Vgl. [Airb96a]

Abbildung 15 veranschaulicht ein Blockdiagramm eines TPIS Systems. Der Druck eines Rades wird mit dem *Pressure Transducer* aufgenommen und zu einer proportionalen Gleichspannung konvertiert. Die Ausgangsspannung des Transducers liegt in einem Bereich von 0 mV bis 100 mV . Dieser Spannungsbereich entspricht einem Druckwertbereich von 0 psi bis 300 psi . Die Ausgangsspannung wird mit Hilfe eines Elektronikmoduls in eine Frequenz von 50 KHz bis 100 KHz transformiert. Durch einen drehenden Transformator zwischen dem Reifen und der Achse kann diese Frequenz an die *TPIS Detection Unit* weitergeleitet werden. Die TPIS Detection Unit digitalisiert das Signal und leitet es über den ARINC 429 Bus weiter an das ECAM System. Eine Warnung wird generiert, wenn der Reifendruck einen bestimmten Grenzwert überschreitet. Andere Räder werden analog behandelt. Alle Räderdrucksignale werden in einer TPIS Detection Unit zusammengeführt. Vgl. [Airb96a]

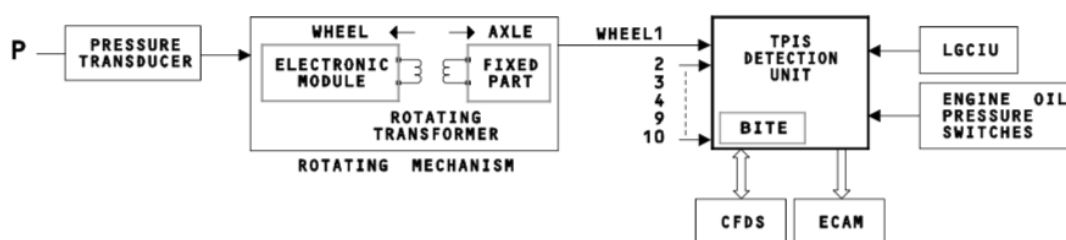


Abbildung 15: Tire Pressure Indicating System. Vgl. [Airb96b]

2.5.2 Brake and Steering Control Unit

Die *Brake and Steering Control Unit* (BSCU) ist ein Teil des Landing Gear Systems. Die BSCU hat nach [Airb96a] drei Aufgaben:

- Die Aufnahme der Bremstemperatur für das ECAM System
- Das Steuern der Bremsen
- Das Lenken des Vorderrades bei der automatischen Landung

Für diese Arbeit ist nur die Bremstemperatur relevant.

¹⁷ Übersichtlich zusammengestellt.

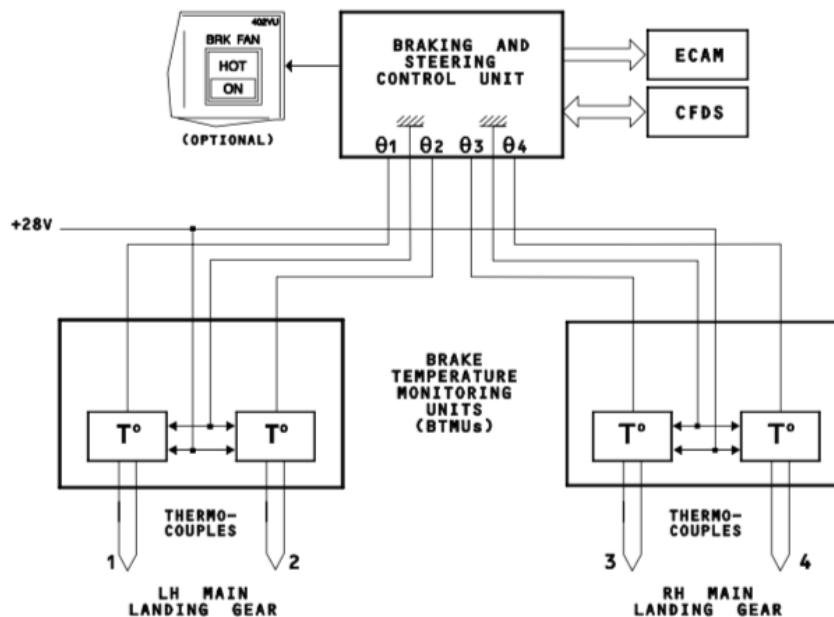


Abbildung 16: Braking and Steering Presentation System. Vgl. [Airb96a]

In Abbildung 16 kann man den Teil der BSCU sehen, der für die Aufnahme und Digitalisierung der Bremstemperatur zuständig ist. Dieser Teil wird Braking and Steering Presentation System (BSPS) genannt. Die Braking Temperature Monitoring Unit (BTMU) ist dafür zuständig, die Werte eines Thermopaars, das direkt an der Radachse befestigt wird, in eine Spannung umzuwandeln. Bei Temperaturen von 0 bis 1000 Grad Celsius werden Spannungen von 1 V bis 9 V erzeugt. Die BSCU nimmt analoge Werte der BTMU auf und wandelt diese in ein digitales Signal um. Eine Warnung wird generiert, wenn die Bremstemperatur einen bestimmten Grenzwert überschreitet. Die BSCU sendet den Temperaturwert über den ARINC 429 Bus an das ECAM System. Vgl. [Airb96a]

2.5.3 Air Data Inertial Reference Unit

Die *Air Data Inertial Reference Unit* (ADIRU) ist dem Navigationssystem zugeordnet. ADIRU besteht aus zwei Systemen: Der *Air Data Reference Unit* (ADR) und der *Inertial Data Reference Unit* (IDR). Beide Systeme nehmen navigationsrelevante Größen auf. Dazu gehören die Gesamtlufttemperatur, die statische Lufttemperatur, die wahre Luftstromgeschwindigkeit, der Anströmwinkel, die Flughöhe, die Fluglage und die Position. Vgl. [Airb96b]

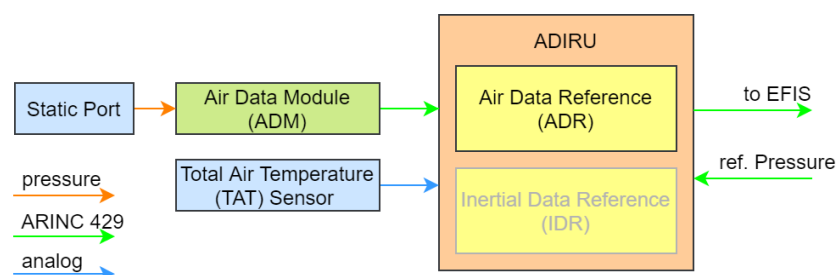


Abbildung 17: ADIRU mit Sensoren. Vgl. [Airb96b]

Abbildung 17 stellt eine ADIRU mit entsprechender Sensorik dar. Der statische Druck wird mit dem *Static Port (SP)* aufgenommen. Er wird durch ein *Air Data Module (ADM)* digitalisiert und über ARINC 429 an die ADIRU gesendet. Der Total Air Temperature (TAT) Sensor liefert eine analoge Spannung, die von der ADIRU direkt digitalisiert wird. Die ADIRU berechnet aus dem Druck und der Temperatur eine Flughöhe und sendet sie an das EFIS System. Für die Flughöhe wird ein Referenzdruck von dem Piloten gewählt und an die ADIRU gesendet. Der Referenzdruck bestimmt, bei welchem Druck der Höhenmesser Null Meter anzeigt. Vgl. [Airb96b]

2.6 Zusammenhang zwischen PFD, SD, DMC, SDAC, ADIRU, TPIS und BSCU

Die Abbildung 18 zeigt den Zusammenhang zwischen allen für diese Arbeit relevanten Flugsystemen. Hierbei ist die im realen System vorhandene Redundanz ausgeblendet.

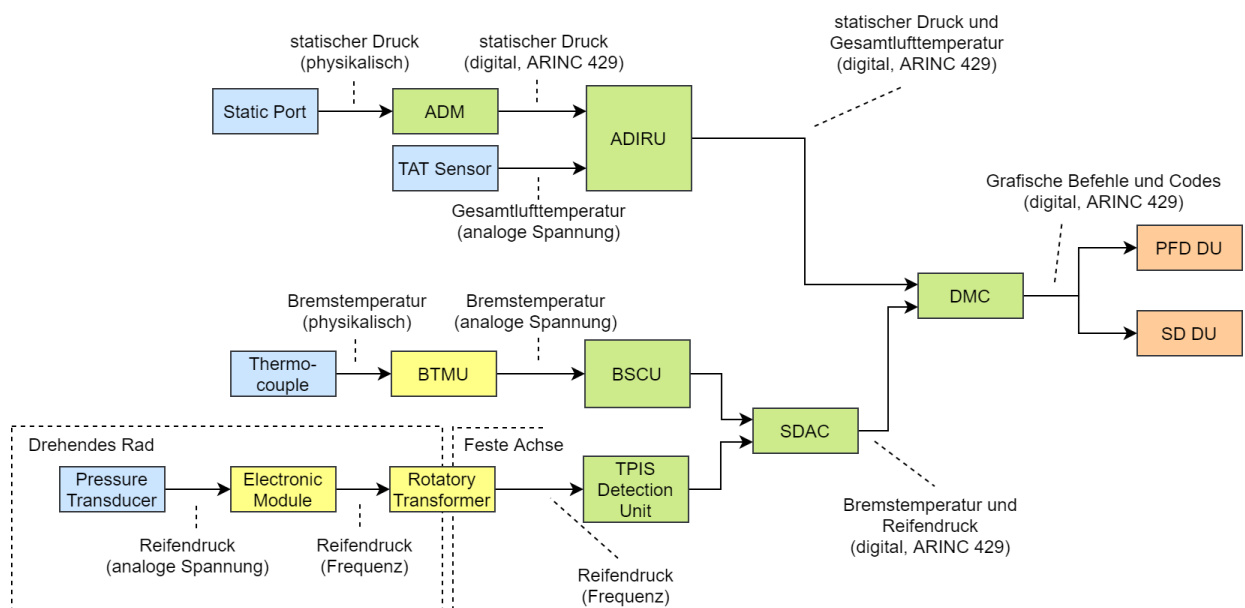


Abbildung 18: Zusammenhang zwischen relevanten Systemen. Vgl. [Airb12, Airb96b, Airb96a]

ADIRU

Der statische Druck wird durch den Static Port aufgenommen. Dieser Wert wird in der ADM in einen digitalen Wert umgewandelt und an die ADIRU gesendet. Die Gesamtlufttemperatur wird mit dem TAT Sensor aufgenommen. Dieser Wert wird als analoge Spannung ausgegeben und von der ADIRU digitalisiert. Die ADIRU sendet den statischen Druck und die Gesamtlufttemperatur an den DMC. Der DMC wandelt die Daten in grafische Befehle und Codes um und sendet sie an die PFD und SD DUs. Der PFD zeigt diese Werte als Flughöhe an und der SD als Gesamtlufttemperatur. Vgl. [Airb12, Airb96b]

BSCU

Die Bremstemperatur wird durch das *Thermocouple* aufgenommen. Die BTMU wandelt die Temperatur in eine Spannung um. Die Spannung wird von der BSCU digitalisiert und über ARINC 429 an den SDAC gesendet. Der SDAC leitet diesen Wert weiter an den DMC. Der DMC bereitet grafische Befehle und Codes vor. Der SD zeigt die Bremstemperatur auf ihrem Bildschirm auf der Wheel Page an. Vgl. [Airb12, Airb96a]

TPIS

Der Pressure Transducer nimmt den Reifendruck auf und wandelt ihn in eine Spannung um. Der Spannungsbereich wird in dem Electronic Module zu einem Frequenzbereich abgebildet. Diese Frequenz kann aus dem drehenden Rad durch einen drehenden Transformator in die Achse übertragen werden. Von der Achse gelangt der Wert in die TPIS Detection Unit. Dort wird der Frequenzwert in einen digitalen Wert umgewandelt. Der digitale Wert wird an den SDAC gesendet. Der SDAC leitet diesen Wert weiter an den DMC. Der DMC bereitet grafische Befehle und Codes vor. Der SD zeigt den Reifendruck auf seinem Bildschirm auf der Wheel Page an. Vgl. [Airb12, Airb96a]

3 Anforderungen

In diesem Teil der Arbeit werden die Anforderungen beschrieben. Diese sind in zwei Kategorien eingeteilt: Anforderungen an den Demonstrator (Abschn. 3.1) und Anforderungen an die Beispielanwendung (Abschn. 3.2). Die Anforderungen an den Demonstrator geben den Rahmen vor, in dem die Flugzeugreferenzarchitektur betrieben wird. Die Anforderungen an die Beispielanwendung spezifizieren ein konkretes Beispiel. Bei letzterem handelt es sich um einen einfachen Messaufbau, der einige wartungsrelevante Messgrößen akquiriert und auf einem Display anzeigt.

Das Vorgehen für die Ermittlung der Anforderungen ist in zwei Phasen aufgeteilt. In Phase 1 werden Anforderungen durch Interviews mit den Projektbeteiligten identifiziert. Ziel ist eine möglichst genaue Wiedergabe der gewünschten Eigenschaften und Parameter. In Phase 2 werden die in Phase 1 identifizierten Anforderungen analysiert und weiter detailliert, so dass ein konkretes technisches System entstehen kann. In diesem Prozess werden sinnvolle Annahmen getroffen und den Projektbeteiligten zwecks Zustimmung vorgestellt.

3.1 Anforderungen an den Demonstrator

Der Demonstrator soll eine mechanisch-elektronische Struktur darstellen, welche luftfahrt-relevante Komponenten in sich tragen kann. Außenflächen sollen das Innere des Demonstrators abdecken. Eine Seite des Demonstrators soll für die Präsentation von Forschungsergebnissen dienen. Dabei soll der Aufbau folgende grundlegende Eigenschaften besitzen: Präsentierbarkeit, Erweiterbarkeit, Autarkie, Verfügbarkeit, Grenzmaße, Grenzgewicht, Personen- und Geräteschutz. Jede Eigenschaft wird einzeln identifiziert und analysiert.

3.1.1 Präsentierbarkeit

Identifizierung

Das Gerät soll auf Messen ausgestellt werden und für eine Zusammenarbeit mit potenziellen Kunden werben. Die Optik des Demonstrators soll also auf den Betrachter präsentabel und animierend wirken und zu einem Gespräch einladen.

Analyse

Um ein konsistentes Aussehen der Plattform zu gewährleisten, wird das Design des Demonstrators an das DLR Logo angelehnt; die Präsentationsflächen im Design erhalten schräge Kanten und weiße, schwarze und graue Farben. Die Optik der tragenden und abdeckenden Komponenten muss bei Belastung konsistent bleiben (niedrige Biegsamkeit) und kleine mechanische Schäden tolerieren können (robust). Die audio-visuelle Darstellung von Anwendungsfällen muss klar und strukturiert sein. Logisch zusammenhängende Komponenten sollten visuell abgegrenzt werden. Für eine höhere visuelle Kohärenz mit eventuell eingesetzten 16:9 Displays kann die Präsentationsfläche auch dasselbe Seitenverhältnis haben.

3.1.2 Erweiterbarkeit

Identifizierung

Das System soll erweiterbar sein. Es soll einen Grundaufbau geben, der mit modularen Komponenten flexibel und einfach erweitert werden kann. Es sollen Werkzeuge verfügbar sein, die hierbei unterstützen.

Analyse

Der Demonstrator muss für den Zweck der Modularität in Anwendungsmodule aufgeteilt werden, die logisch zusammenhängenden Komponenten auch zusammen darstellen. Die Komponenten können aus Mechanik, Elektronik und Software bestehen. Die mechanischen Komponenten sollen schraublos befestigt werden und die elektrischen Verbindungen der Komponenten ohne Lötarbeiten durchgeführt werden können. Komponenten, die kommunikationsfähig sind, müssen nachrichtenbasiert Daten untereinander austauschen können. Die Kommunikationsverbindung muss programm-technisch abstrahiert werden. Werkzeuge für die Entwicklung, Diagnose und den Test der Kommunikation werden zur Verfügung gestellt. Für elektronische Komponenten mit Rechentechnik stehen eine Hardwareabstrahierung und Treiber zur Verfügung. Interpretierte Programmiersprachen sollen von der eingesetzten Software unterstützt werden.

Da die Verwendung von kompatiblen Geräten und Lösungen auch kompatible Technologien voraussetzt, werden durch die Wahl konkreter Technologien wegen Technologieabhängigkeiten stets Nachfolgeanforderungen definiert. Die Darstellung solcher Anforderungen erfolgt in Tabelle 25.

3.1.3 Autarkie

Identifizierung

Das System soll im Sinne der Präsentation und Entwicklung autark sein. Der Demonstrator soll in einer definierten Zeit selbstständig initialisieren können und aus einer Stromquelle gespeist werden. Die Versorgungsspannung der Komponenten ist so zu wählen, dass Geräte wie Mini-PC's, Bildschirme und SPS in unmittelbarer Zukunft integriert werden können.

Analyse

Es muss ein 230 VAC Kaltgeräteanschluss sowie ein Hauptschalter für die einfache Inbetriebnahme integriert werden. Der Schalter aktiviert dann die 230 VAC Spannung, die von einem Netzteil in eine Gleichspannung umgewandelt wird. Die in „Identifizierung“ genannten Geräte benötigen im Schnitt eine geringere Spannung als 24 VDC. Diese Spannung stellt die Nennspannung des Wandlers dar. Die Gleichspannung wird an alle Anwendungskomponenten durch ein standardisiertes Energieverteilungssystem verteilt. Es sollte möglich sein, die 24 VDC Gleichspannung in eine kleinere Gleichspannung umwandeln zu können um auch Geräte zu versorgen, die weniger als 24 VDC benötigen. Hierbei muss die Stromwelligkeit des Endverbrauchers beachtet werden. Da nicht bekannt ist wie viele Komponenten mit welchen Strombedarfen zukünftig zu integrieren sind, sollte bei der Auswahl der Komponenten die Möglichkeit einer geringen Stromaufnahme berücksichtigt werden. Bei der Verteilung sollte darauf geachtet werden, dass die Stromauslastung der Kabel durch einen entsprechenden Netzwerktyp minimiert wird.

Für die Stromgrenze des Stromversorgungsgerätes muss eine sinnvolle Annahme getroffen werden, die einerseits dem Strombedarf des jetzigen Projektes gerecht wird und andererseits auch den Bedarfen zukünftiger Projekte. Es wird angenommen, dass im nachfolgenden Projekt alle in 2.4 und in 2.5 aufgeführten Flugsysteme benötigt werden. Die Redundanz der Computer wird dabei nicht mitgerechnet. Laut 2.4 und in 2.5 müssen dann in etwa 10 Avionics Computer versorgt werden können. Es wird ein Puffer von 5 weiteren Komponenten dieser Art angenommen. Somit muss die Stromversorgung 15 Avionics Computer versorgen können. Zusätzlich ist an einen Puffer für weitere Elektronik zu denken. Der genaue Stromgrenzwert wird auf Grundlage der gewählten Geräte, die als Avionics Computer dienen, in Kapitel 4 dargelegt.

Für den Demonstrator wird auch eine zeitliche Vorgabe gesetzt: Das Initialisieren aller elektronischen und programm-technischen Komponenten muss in weniger als 2 Minuten abgeschlossen sein. So kann der Demonstrator bei einer ungeplanten Vorführung schnell genug in Betrieb genommen werden.

3.1.4 Verfügbarkeit

Identifizierung

Es sollen nach Möglichkeit bereits verfügbare, nachhaltige und wirtschaftliche Lösungen gewählt werden.

Analyse

Die zu integrierenden Komponenten sollen gut dokumentiert sein und genug Anwendungsbeispiele mit sich bringen. Der Hersteller und/oder die Community sollen eine Langzeitunterstützung anbieten, die auf eine Weiterentwicklung der Hardware/Software und ihren Erwerb in unmittelbarer Zukunft abzielt. Im Vergleich zu einer spezialisierten Luftfahrtlösung müssen die Gesamtkosten für den Aufbau und Komponenten gering gehalten werden.

3.1.5 Grenzmaße

Identifizierung

Die Abmaße des Aufbaus betragen in Breite und Höhe in etwa jeweils 2 m. Die Tiefe soll sinnvoll gewählt werden.

Analyse

Die Tiefe des Demonstrators richtet sich nach der gewählten Elektronik und den möglichen zukünftigen Bedarfen. Es wird davon ausgegangen, dass im Demonstrator Komponenten eingesetzt werden, die in 3.1.3 genannt wurden. Zusätzlich sollen auch Router, Netzteile, Einplatinencomputer, Mikrokontroller sowie Geräte mit ähnlichen Maßen dort Platz haben. Geräte mit größeren Abmessungen sollten außerhalb des Demonstratoraufbaus betrieben werden. Die erwähnten Komponenten haben im Schnitt eine geringere Tiefe als 150 mm. Zusammen mit einem Puffer von 50 mm für die Gesamtlänge einer Grundflächenseite der Trageprofile soll die Gesamttiefe des Demonstrators 200 mm betragen. Da nicht bekannt ist wie viele Komponenten mit welchen Maßen zukünftig integriert werden, sollte die Wahl der Komponenten möglichst kompakt sein. Viele Türdurchgänge haben eine Höhe von exakt 2 m, somit kann die Höhe des Demonstrators niedriger ausgelegt werden. Der Demonstrator kann dadurch bei Bedarf aufrecht durch den Türrahmen durchgetragen werden.

3.1.6 Grenzgewicht

Identifizierung

Das Grenzgewicht des gesamten Demonstrators soll maximal 100 kg betragen.

Analyse

Es sollte genügend Gewichtspuffer für zukünftig einzusetzende Bauteile einkalkuliert werden. Bei den in 3.1.5 erwähnten Komponenten wird in Summe mit einem Gewicht von 30 kg kalkuliert. Für das Trägersystem mit den Außenflächen bleiben damit 70 kg. Da zu diesem Zeitpunkt nicht bekannt ist wie viele Anwendungskomponenten und mit welchem Gewicht zukünftig integriert werden, sollte die Wahl bei Möglichkeit leicht sein.

3.1.7 Personen- und Geräteschutz

Identifizierung

Der Demonstrator ist eine Laborversuchs- und gleichzeitig eine Präsentationsplattform. Bei Versuchsdurchführungen und öffentlichen Ausstellungen muss der Personen- und Geräteschutz gewährleistet werden.

Analyse

Der Demonstrator muss durch versehentliches Anlehnen Kippstabilität beibehalten. Elektrisch leitende Bauteile sind zu erden. Die Betriebsgleichspannung der Komponenten muss unter 60 VDC bleiben, um das offene Arbeiten an spannungsaktiven Bauteilen zu ermöglichen [Bghm12]. Der Kabelquerschnitt für die Verteilung der Komponentenspannung muss die ausgelegte Stromgrenze tolerieren können [Kabe00]. Für den Geräte- und Kabelschutz sind Sicherungen einzuplanen. Da sich Sicherungen und Kabelquerschnitt nach der Stromgrenze des Spannungswandlers richten, werden diese Parameter erst in Kapitel 4 betrachtet.

3.2 Anforderungen an die Beispielanwendung

Die Anforderungen an die Beispielanwendung erweitern die Spezifikation des Demonstrators. Die Anwendung besitzt dabei Anforderungen, welche nicht strikt voneinander trennbare Abhängigkeiten aufweisen. Dementsprechend werden die Anforderungen zunächst als ein Ganzes identifiziert und anschließend analysiert.

3.2.1 Identifizierung

Der Kontext

Die Beispielanwendung soll innerhalb des Demonstrators umgesetzt werden. Es sollen wartungsrelevante Messungen durchgeführt und auf Displays angezeigt werden. Die Messungen sollen einen Erfahrungswert für den potentiellen Kunden und für die Luftfahrtforscher bringen. Dieser Erfahrungswert erlaubt es, das Potential für die weiterführende Entwicklung einzuschätzen. So kann entschieden werden, ob eine Zusammenarbeit mit anderen Firmen möglich ist. Aus diesem Grund sind reale Erwartungswerte, Genauigkeiten und Messbereiche eines Flugmesssystems zweitrangig, wichtiger ist der Präsentationsfaktor. Die durchzuführenden Messungen sollten dem Benutzer performant¹⁸ und feinkörnig¹⁹ erscheinen. Die Auslegung der Messgenauigkeiten und der Messbereiche sollten sich grundsätzlich an eine Labor- bzw. Ausstellungsumgebung richten.

Für die Durchführung einer Beispielanwendung werden vier wartungsrelevante Messgrößen eingeführt:

- Bremsentemperatur der Räder
- Druck der Räder
- Gesamtlufttemperatur
- statischer Druck

Diese Messgrößen sollen von den folgenden nachgebildeten Flugsysteme aufgenommen, verarbeitet und angezeigt werden:

- Display Unit (DU),
- Display Management Computer (DMC),

¹⁸ Schnelle Reaktion

¹⁹ Viele Abstufungen

- System Data Acquisition Concentrator (SDAC),
- Tire Pressure Indicating System (TPIS),
- Brake and Steering Control Unit (BSCU),
- Air Data Inertial Reference Unit (ADIRU).

Die Komponenten

Der Projektleiter definiert fünf grundlegende Komponentengruppen des Modells auf der Grundlage der realen Avionikstruktur:

- Wartungsrelevante Sensoren,
- Avionics Computer (AC),
- Data Modules (DM),
- Display Units (DU) und
- lokale Displays (LD).

Ein Data Module soll eine Verallgemeinerung des Air Data Modules darstellen und ist für die Akquirierung der Sensordaten verantwortlich. Ein lokales Display soll der Überprüfung von unmittelbar akquirierten Sensordaten dienen. Innerhalb eines Avionics Computers kann der grundlegenden Idee der Trennung in ein IO Interface und eine Processing Unit (PU) gefolgt werden (vgl. 2.4.1). Die Trennung kann sowohl in HW als auch in SW stattfinden.

Der statische Drucksensor, der Raddrucksensor, der Gesamtlufttemperatursensor und der Bremstemperatursensor bilden die Sensorgruppe. ADIRU, BSCU, TPIS, SDAC, DMC bilden die Avionics Computer. Das Tire Pressure Data Module (TPDM), Brake Temperature Data Module (BTDM) und Air Data Module (ADM) bilden die Gruppe der DM's. Das System Display (SD) ist eine DU.

Display Units

Für die zukünftigen Erweiterungen des Cockpits sollten vier Displays als Display Units eingebaut werden.

Die Kommunikation

Die Avionics Computer sollen über das ARINC 429 Bussystem kommunizieren. Wenn es nicht möglich ist die reale ARINC 429 Kommunikationstechnologie zu verwenden, dann soll die grundlegende Verbindungs- und Adressierungsfunktion von ARINC 429 nachgebildet werden. Dies kann sowohl mit SW als auch mit HW realisiert werden. Die Kommunikation zwischen den DMs und Avionics Computern kann über ein einfaches Punkt-zu-Punkt²⁰ Kommunikationsprotokoll stattfinden.

Der Zusammenhang

Abbildung 19 zeigt die geforderte Flugreferenzstruktur, die nachgebildet werden muss. Die Struktur ist an die realen Verbindungen in dem Avionics Compartment angelehnt, wobei einige Modifikationen durch den Projektleiter definiert wurden (vgl. Abbildung 18). TPIS misst den Raddruck über einen TPDM und sendet ihn an den SDAC. Das BSCU misst die Bremstemperatur über einen BTDM und sendet auch an den SDAC. Die ADIRU soll über einen ADM Gesamtlufttemperatur und den statischen Druck messen. Beide Werte werden dann an den DMC gesendet. BTDM, TPDM, ADM zeigen ihre Messwerte auf lokalen Displays an. Die Daten des SDAC werden an den DMC gesendet. Der DMC leitet die Daten der ADIRU und des SDAC weiter

²⁰ Direkte Verbindung zwischen zwei Kommunikationsteilnehmern

an das SD. Das SD soll in einer freien Form den Gesamtluftdruck, den statischen Druck, den Raddruck und die Bremstemperatur anzeigen. Die Präsentation der Messungen soll für den Betrachter visuell klar strukturiert sein.

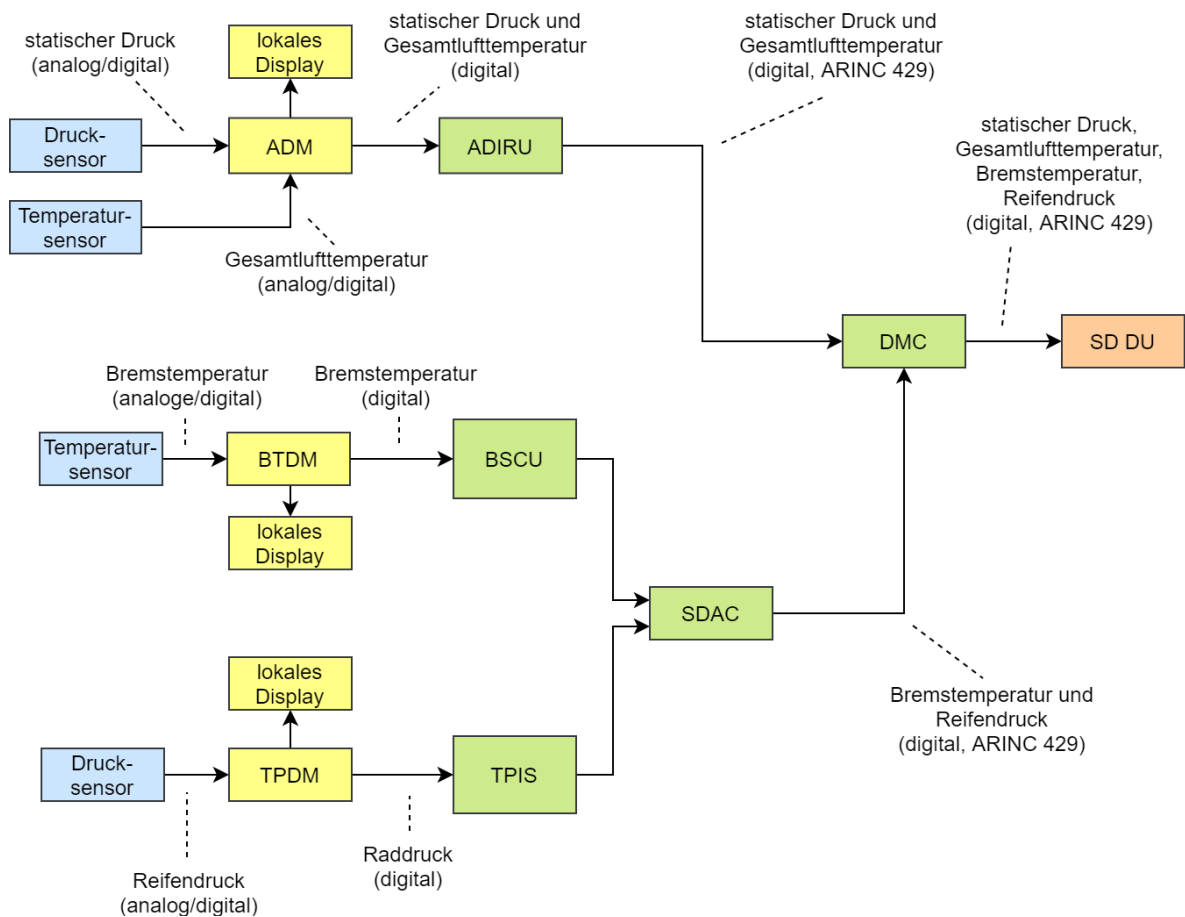


Abbildung 19: Verbindungsstruktur der geforderten Flugsysteme

3.2.2 Analyse

Auf Grundlage des Strukturmodells in Abbildung 19 werden für die Komponentengruppen sinnvolle Annahmen getroffen. Avionics Computer, Data Modules, Display Units und lokale Displays sind allgemein zu betrachten, da sich die einzelnen Instanzen nicht unterscheiden. Die Sensoren werden separat betrachtet, da jeder Sensor über andere Anwendungsaspekte verfügt.

Gesamtlufttemperatursensor

Der Demonstrator soll in einem geschlossenen, gut belüfteten und an kalten Tagen beheizten Raum betrieben werden. Da sich der Demonstrator im Stillstand befindet, entspricht die Gesamtlufttemperatur der Umgebungstemperatur. Die zu erwartende Umgebungstemperatur wird durchschnittlich 20 Grad betragen. Die Messung soll mindestens in einem Bereich von 0 bis 50 Grad erfolgen um Wetterschwankungen entsprechend zu berücksichtigen. Für das Gefühl der Feinkörnigkeit soll die Messgenauigkeit mindestens 0,5 Grad betragen.

Statischer Drucksensor

Identische Raumbedingungen gelten für den statischen Druck, wobei der statische Druck dem aktuellen Umgebungsdruck entspricht. Dieser variiert abhängig von dem örtlichen Wetter. Der zu erwartende Umgebungsdruck liegt im Schnitt bei 1013,25 mbar. Die Messung sollte mindestens in einem Bereich von 950 - 1060 mbar durchgeführt werden, um die üblichen Wetter-

schwankungen abbilden zu können. Die Messgenauigkeit sollte mindestens bei 0,5 mbar liegen, was auch in diesem Fall ein ausreichendes Gefühl für die Feinkörnigkeit der Messung erzeugen würde.

Bremstemperatursensor

Die Bremstemperatur soll mit einem Bunsenbrenner simuliert und das Temperaturelement mit einer metallischen Hülle ummantelt werden, um die Flamme des Bunsenbrenners gegenhalten zu können. Das zu verwendende Gas sollte keine Spuren auf dem Sensorelement hinterlassen. Der Messbereich des Temperaturelements sollte 0 - 200 Grad sein, um genügend Spielraum bei der Erhitzung mit dem Bunsenbrenner zu haben. Der Ruhewert sollte der Raumtemperatur entsprechen. Die Messgenauigkeit sollte mindestens bei 1 Grad liegen.

Raddrucksensor

Der Raddruck sollte mit einer Pumpe simuliert werden können. Der Messbereich des Drucksensors sollte 0 – 4 bar sein, was auch hier genug Spielraum für die Druckänderung gibt. Der Ruhewert liegt bei 0 bar und entspricht dem Umgebungsdruck. Die Messgenauigkeit sollte mindestens bei 10 mbar liegen. Da ein Druck sich sehr schnell verändern kann, sollte die Messung auf dem Display auch schnell genug aktualisiert werden können.

Avionics Computer

Die primäre Aufgabe der Avionics Computer ist das Verarbeiten von Daten und das Versenden an andere Avionics Computer. Bei einem realen Avionics Computer handelt es sich um einen Mikroprozessor mit einem sehr spezifischen Programm, das stets für genau eine sehr spezifische Aufgabe entwickelt und eingesetzt wird. Die Softwareentwicklung wird hardwarenah durchgeführt. Im Gegensatz dazu müssen die Avionics Computer des Demonstrators mit einem gängigen Betriebssystem ausgestattet werden. Dadurch lassen sich Entwicklungsaufwand und Entwicklungszeit reduzieren. Ein entsprechendes Betriebssystem bietet eine niedrige Einstiegsschwelle für die Forscher, die eine hardwarenahe Programmierung nicht kennen.

Es wird angenommen, dass der Avionics Computer für die Anwendungsbereiche Hochfahren, Kompilieren und Normalbetrieb eingesetzt wird. In allen drei Fällen sollte der Computer eine ausreichend schnelle Performance anbieten, um einen effizienten Entwicklungsprozess zu garantieren.

Betriebssystem für die Avionics Computer

Das Betriebssystem sollte über ein großes Repositorium an bereits vorhandenen Paketen und Bibliotheken sowie eine grafische Oberfläche für die einfache Handhabung des Rechners verfügen.

Data Modules

Die Aufgabe der Data Modules ist das Sammeln von Sensordaten, die an die Avionics Computer versendet werden. Die Data Modules müssen gängige Kommunikations- und Wandlungstechnologien besitzen, unter anderem UART, I2C, SPI und Analog-Digital-Wandler. Die DMs sollten es ermöglichen, echtzeitfähige und deterministische Anwendungen entwickeln zu können. Deshalb muss der Data Module als Computertyp ein Mikrokontroller sein.

Display Unit

Die Display Unit wird im Rahmen dieser Bachelorarbeit für das Anzeigen von vier wartungsrelevanten Messgrößen verwendet, die mit arbiträren Grafiken ausgestattet werden sollen. Displaygröße und Auflösung sollten nicht zu klein gewählt werden, um Spielraum bei der

Gestaltung zu haben. Mit den Displays wird die grafische Umgebung eines Betriebssystems angezeigt. Dadurch können gängige grafische Frameworks verwendet werden, ohne eine Beschränkung durch die HW zu erfahren. Da die obere Schranke für die Displaygröße nicht vorgegeben wurde, wird die reale Display Unit aus dem Cockpit als Referenz verwendet. Sie hat ein Seitenverhältnis von 1:1 und die Länge einer Displayseite von 7,25 Zoll. Da die meisten modernen Displays über kein quadratisches Seitenverhältnis verfügen, sollte nur die längste Seite etwa sieben Zoll betragen. Für die Abbildung umfangreicher Details kann das Display eine Auflösung über 1280 x 720 besitzen.

Lokales Display

Die lokalen Displays dienen der Überprüfung der unmittelbar gemessenen Sensorwerte innerhalb eines Data Modules. Die lokalen Displays sollen nicht so präsent sein wie die Display Units. Somit soll die Displaygröße bei Möglichkeit klein sein aber ausreichend Platz für die Abbildung der Messwerte haben. Für die Anzeige reichen zwei Farben aus.

Kommunikation über ARINC 429

Die Integration der realen ARINC 429 Kommunikationstechnik ist zeitaufwendig und nicht wirtschaftlich. Die bereits erhältlichen ARINC 429 Adapter können von dem Projektbudget nicht getragen werden. Eine eigene Entwicklung der ARINC 429 Adapter ist, bedingt durch Entwicklungskosten und lange Lieferzeiten, nicht realisierbar. Deswegen muss für das Kommunikationssystem ein Ersatz gefunden werden, der die zwei geforderten ARINC 429 Eigenschaften nachbildet: Die Verbindung über die 1-n Topologie und Adressierung über Labels.

Die Anwendung benötigt die aufgenommenen Messwerte primär für die visuelle Vorführung. Daher sollen sich die Latenzen der Übertragung von einem Sensor zu einem Display an die Aktualisierungsrate der Zahlenwerte auf den Displays richten. Die Aktualisierungsrate hängt mit dem menschlichen Faktor zusammen. Ab 16 Bilder pro Sekunde kann der Mensch einzelne Bilder einer Bildreihenfolge nicht mehr unterscheiden. Er sieht stattdessen eine durchgehende Bewegung. Vgl. [Neum14]. Dasselbe gilt auch für Zahlenwerte, da eine Erneuerung eines Zahlenwertes auch als eine Erneuerung eines Bildes betrachtet werden kann. Das Hirn wird die einzelnen niederwertigen Ziffern einer Dezimalzahl bedingt durch zu schnelle Veränderungen nicht unterscheiden können [Arin04]. Daher soll das Aktualisierungsintervall der Zahlenwerte auf den Displays nicht kürzer als $1/16 \text{ BpS}^{21} = 62,5 \text{ ms}$ sein. Die Latenzzeit für die Kommunikation zwischen Quelle und Ziel soll daher unter 62,5 ms liegen.

Middleware für die Avionics Computer

Die Middleware dient der Kommunikationsabstrahierung unter den Kommunikationsteilnehmern. Es muss möglich sein, die Kommunikationsverbindung mit der Middleware in drei Fällen abstrahieren zu können:

- Bei der Kommunikation zwischen den Avionics Computern.
- Bei der Kommunikation zwischen den Softwaremodulen der Avionics Computer.
- Bei der Kommunikation zwischen den Avionics Computern und den Data Modules.

Die Kommunikationsteilnehmer müssen sich über vordefinierte oder selbst erstellte Nachrichtentypen mit Daten austauschen können. Die Middleware muss es erlauben, die grafischen Frameworks auf eine einfache Art und Weise in die Kommunikationskette

²¹ Bilder pro Sekunde

einzubinden. Es soll ein Paketmanager vorhanden sein und ein großes Repositorium an modularen Softwarepaketen angeboten werden.

3.3 Tabellarische Zusammenfassung der Anforderungen

Alle gesammelten Anforderungen werden in drei Tabellen gesammelt:

- Anforderungen an den Demonstrator (Tabelle 1).
- Anforderungen an die Beispielanwendung (Tabelle 2).
- Nachfolgeanforderungen (Tabelle 3).

Die Tabellen bilden fünf Spalten ab: Kategorie, Nummer, Beschreibung, Priorität und Erfüllt. Kategorie sagt aus, zu welcher Gerätegruppe oder Eigenschaft die Anforderung gehört. Die Nummern ermöglichen eine einfache Referenzierung in nachfolgenden Kapiteln dieser Arbeit. Die Beschreibung definiert die eigentliche Anforderung. Die Priorität sagt aus, wie wichtig es ist, diese Anforderung zu erfüllen. In der Spalte „Erfüllt“ wird die Information vorgezogen, ob die Realisierung der entsprechenden Anforderung nachgekommen ist.

Um die Prioritäten zu bestimmen, wird jede Anforderung den folgenden Fragen unterzogen:

- Betrifft die Anforderung die Personensicherheit?
- Ist die Anforderung für die grundlegende Funktionalität des Demonstrators essentiell?
- Ist die Anforderung durch die Projektleitung vorgegeben?

Wenn eine Nichterfüllung der Anforderung zur Gefährdung von Personen führen kann, ist ihre Priorität hoch. Explizit mit der Projektleitung vereinbarte Anforderungen haben zumindest die Priorität „mittel“. Wenn der Demonstrator ohne die Anforderung nicht mehr seinem definierten Grundwesen entspricht oder die Stabilität seiner Hard- oder Softwarekomponenten gefährdet ist, wird die Priorität als hoch eingestuft. Anforderungen, die nur kleine, insignifikante technische oder optische Verbesserungen mit sich bringen, haben eine niedrige Priorität.

Tabelle 1: Zusammenfassung der allgemeinen Anforderungen

Kategorie	Nr.	Beschreibung	Priorität	Erfüllt
Präsentierbarkeit	1	Die Präsentationselemente sollen optische Ästhetik aufweisen	mittel	Ja
	2	Das Design der Präsentationsflächen kann schwarze, weiße, graue Farben und schräge Kanten aufweisen	niedrig	Ja
	3	Die tragenden und abdeckenden Komponenten müssen niedrige Biegsamkeit aufweisen	hoch	Ja
	4	Die tragenden und abdeckenden Komponenten müssen robust gegen mechanische Schäden sein	hoch	Ja
	5	Die audio-visuelle Darstellung des Anwendungsfalls muss klar und strukturiert sein	hoch	Ja
	6	Logisch zusammenhängende Komponenten sollen visuell abgegrenzt werden	mittel	Ja
	7	Die Präsentationsfläche kann ein Seitenverhältnis von 16:9 haben	niedrig	Ja

Erweiterbarkeit	8	Der Demonstrator muss in Anwendungsmodule aufgeteilt werden	hoch	Ja
	9	Mechanische Komponenten sollen schraublos befestigt werden	mittel	Ja
	10	Elektronische Komponenten sollen ohne Lötarbeiten verbunden werden	mittel	Ja
	11	Die Kommunikationsteilnehmer müssen über Nachrichtenpakete Daten austauschen können	hoch	Ja
	12	Die Kommunikation muss programm-technisch abstrahiert werden	hoch	Ja
	13	Es sollen Werkzeuge für Entwicklung, Diagnose und Test der Kommunikation zur Verfügung stehen	mittel	Ja
	14	Die Hardwareabstrahierung/Treiber sollen zur Verfügung stehen	mittel	Ja
	15	Die Komponenten sollen wenig Aufwand bei Entwicklung, Implementierung und Anwendung benötigen	mittel	Ja
	16	Interpretierte Programmiersprachen sollen unterstützt werden	mittel	Ja
	17	Es sollen möglichst kompatible Geräte und Lösungen eingesetzt werden	mittel	Ja
Autarkie	18	Es muss ein 230 VAC Kaltgeräteanschluss integriert werden, ein Hauptschalter muss die Spannung aktivieren	hoch	Ja
	19	230 VAC muss in eine Nenngleichspannung von 24 VDC umgewandelt werden	hoch	Ja
	20	24 VDC soll in eine niedrige Gleichspannung umgewandelt werden	mittel	Ja
	21	Die Stromwelligkeit der Endverbraucher muss beachtet werden	hoch	Ja
	22	Die Komponenten sollen niedrige Strombedarfe aufweisen	mittel	Ja
	23	Der Strom muss durch ein standardisiertes Energieverteilungssystem verteilt werden	hoch	Ja
	24	Die Stromauslastung der Kabel des Energieverteilungssystems soll niedrig gehalten werden	mittel	Ja
	25	Die Stromgrenze des Spannungswandlers muss so ausgelegt sein, dass 15 Avionics Computer und DU's zusammen mit weiterer Elektronik von der Gleichspannungsquelle getragen werden können	hoch	Ja
	26	Der Demonstrator muss selbstständig unter 2 Minuten initialisieren	hoch	Ja
Verfügbarkeit	27	Die Komponenten sollen gut dokumentiert sein und es soll Anwendungsbeispiele geben	mittel	Ja

	28	Die Komponenten sollen die Langzeitunterstützung des Herstellers und/oder der Community haben	mittel	Ja
	29	Die Gesamtkosten müssen gering gehalten werden	hoch	Ja
Grenzmaße	30	Die Maße des gesamten Demonstratoraufbaus sollen höchstens 2 m Breite, 2 m Höhe und 0,2 m Tiefe haben	mittel	Ja
	31	Die Höhe kann so angepasst werden, dass der Demonstrator aufrecht durch einen Türrahmen getragen werden kann	niedrig	Ja
	32	Die Tiefe der Komponenten muss unter 150 mm bleiben	hoch	Ja
	33	Die Gesamtlänge der Grundflächenseiten der Trageelemente soll unter 50 mm sein	mittel	Ja
	34	Die Komponenten sollen möglichst kompakt sein	mittel	Ja
Grenzwert	35	Das Gewicht der Abdeckungen und des tragenden Profils muss unter 70 kg bleiben	hoch	Ja
	36	Das Gesamtgewicht des Demonstrators muss unter 100 kg bleiben	hoch	Ja
	37	Die Komponenten sollen ein möglichst niedriges Gewicht haben	mittel	Ja
Personenschutz und Sicherheit	38	Bei versehentlichem Anlehnen muss der Aufbau kippstabil bleiben	hoch	Ja
	39	Elektrisch leitende Bauteile müssen geerdet sein	hoch	Ja
	40	Die Betriebsgleichspannung der Komponenten muss unter 60 VDC bleiben	hoch	Ja
	41	Für den Geräte- und Kabelschutz müssen die Sicherungen entsprechend der Stromgrenze ausgelegt sein	hoch	Ja
	42	Der Kabelquerschnitt muss die benötigten Ströme tragen können	hoch	Ja

Tabelle 2: Zusammenfassung der Anwendungsanforderungen

Kategorie	Nr.	Beschreibung	Priorität	Erfüllt
Verbindungsstruktur der nachgebildeten Flugsysteme	43	Die Flugsysteme müssen nach der dargestellten Struktur in Abbildung 19 verbunden werden	hoch	Ja
Gesamtlufttemperatursensor	44	Der Messbereich soll 0 - 50 Grad sein	mittel	Ja
	45	Die Messgenauigkeit soll mindestens 0,5 Grad sein	mittel	Ja

Statischer Drucksensor	46	Der Messbereich soll 950 - 1060 mbar sein	mittel	Ja
	47	Die Messgenauigkeit soll mindestens 0,5 mbar sein	mittel	Ja
Bremsentemperatur-sensor	48	Der Messbereich soll 0 - 200 Grad sein	mittel	Ja
	49	Die Messgenauigkeit soll mindestens 1 Grad sein	mittel	Ja
Raddrucksensor	50	Der Messbereich soll 0 - 4 bar sein	mittel	Ja
	51	Die Messgenauigkeit soll mindestens 10 mbar sein	mittel	Ja
Avionics Computer	52	Es muss ein Betriebssystem eingesetzt werden	hoch	Ja
	53	Beim Hochfahren, Kompilieren und im Normalbetrieb soll eine ausreichende Performanz angeboten werden	mittel	Ja
Betriebssystem für die Avionics Computer	54	Es soll ein Paketmanager vorhanden sein und ein großes Repositorium an Softwarepaketen angeboten werden	mittel	Ja
	55	Es muss eine grafische Oberfläche vorhanden sein	hoch	Ja
Data Modules	56	Der Computertyp muss ein Mikrokontroller sein	hoch	Ja
	57	Es muss die Unterstützung von UART, I2C, SPI und einem Analog-Digital Wandler geben	hoch	Ja
Display Units	58	Es soll die grafische Oberfläche des Betriebssystems angezeigt werden können	mittel	Ja
	59	Die längste Seite des Bildschirms soll unter 7 Zoll sein	mittel	Ja
	60	Die Bildschirmauflösung kann über 1280 x 720 liegen	niedrig	Nein
	61	Es sollen vier Display Units eingebaut werden	mittel	Ja
Lokale Displays	62	Das Display kann zwei Farben anzeigen	niedrig	Ja
Kommunikation über ARINC 429	63	Es muss die Verbindungs- und Adressierungsfunktion nachgebildet werden	hoch	Ja
	64	Die Latenzen sollen unter 62,5 ms sein	mittel	Ja
Middleware für die Avionics Computer	65	Es muss die Abstrahierung der Kommunikation zwischen den Avionics Computern ermöglicht werden	hoch	Ja
	66	Es muss die Abstrahierung der Kommunikation innerhalb der Avionics Computer ermöglicht werden	hoch	Ja

	67	Es muss die Abstrahierung der Kommunikation zwischen einem Avionics Computer und einem Data Modul ermöglicht werden	hoch	Ja
	68	Es müssen vordefinierte oder benutzerdefinierte Nachrichtentypen verwendet werden können	hoch	Ja
	69	Es müssen grafische Frameworks in die Kommunikation integriert werden können	hoch	Ja
	70	Es soll ein Paketmanager vorhanden sein und ein großes Repository an Softwarepaketen angeboten werden	mittel	Ja

Tabelle 3: Zusammenfassung der Nachfolgeanforderungen

Kategorie	Nr.	Beschreibung	Priorität	Erfüllt
Kompatibilität mit ROS	71	Die Komponenten sollen mit ROS kompatibel sein	mittel	Ja
Kompatibilität mit PI 3B+	72	Die Komponenten sollen mit PI 3B+ kompatibel sein	mittel	Ja
Hutschienenanschluss	73	Die Komponenten sollen über einen Hutschienenanschluss verfügen	mittel	Ja

4 Konzeption

In diesem Kapitel werden auf Grundlage der Anforderungen konzeptionelle Entscheidungen zu der Wahl von technischen Komponenten getroffen.

Abbildung 20 zeigt das grundlegende Strukturkonzept der Systeme des Demonstrators. Das mechanische Trägersystem und die Stromversorgung stellen den eigentlichen Demonstratoraufbau dar. Das Kommunikationssystem und die Komponenten bilden die Beispielanwendung. Die Beispielanwendung verwendet die Infrastruktur des Demonstrators. Die elektronischen Komponenten der Beispielanwendung werden physisch durch das mechanische Trägersystem gehalten und durch das Stromversorgungssystem mit Strom versorgt.

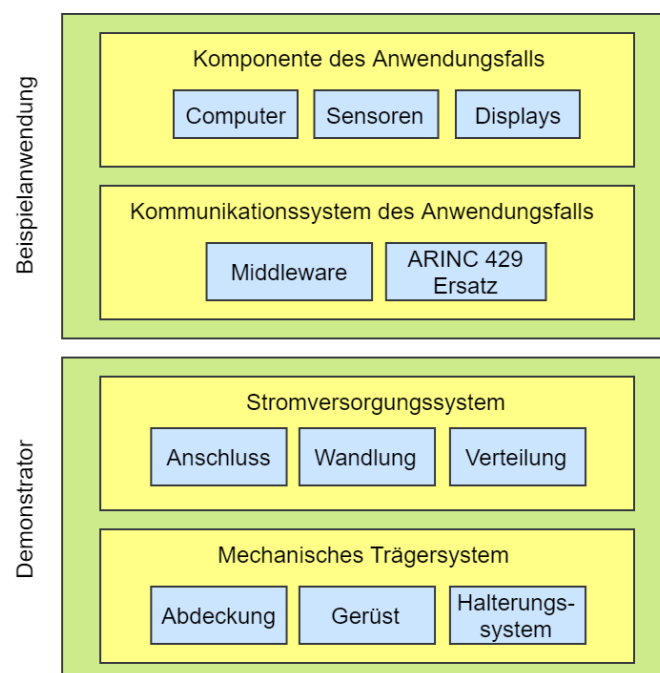


Abbildung 20: Übersicht über das grundlegende Strukturkonzept

Einige eingesetzte Technologien sind voneinander abhängig. Entsprechend wird in diesem Kapitel eine bestimmte Reihenfolge der Konzeption eingehalten. Die Technologien, die als Grundlage für andere Technologien dienen, werden zuerst behandelt. Die in Abbildung 20 erwähnten Komponentengruppen werden in folgender Reihenfolge bearbeitet:

- Mechanisches Trägersystem (Abschn. 4.1)
- Kommunikationssystem des Anwendungsfalls (Abschn. 4.2)
- Komponenten des Anwendungsfalls (Abschn. 4.3)
- Stromversorgungssystem (Abschn. 4.4)

4.1 Mechanisches Trägersystem

In diesem Unterkapitel werden Materialien für den mechanischen Aufbau gewählt.

4.1.1 Abdeckungen

Die Abdeckungen haben zwei Funktionen: Einerseits hängt von ihnen die Optik ab, andererseits sollten sie visualisierende Komponenten tragen können; unter anderem Bildschirme und die Elektronik, die für die Präsentation von Forschungsergebnissen benötigt werden. Verglichen werden zwei Materialien mit den Stärken 3 mm und 5 mm:

- Weiß lackierte MDF²² Holzplatte.
- Weißes Acrylglas vom Typ XT.

Tabelle 4: Vergleich von Abdeckungen

Nr.	Anforderung	MDF 3 mm	MDF 5 mm	Acrylglas XT 3 mm	Acrylglas XT 5 mm
27	Gut dokumentiert / es gibt Anwendungsbeispiele	+	+	+	+
28	Langzeitunterstützung des Herstellers / der Community	+	+	+	+
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+	+	++	++
3	Niedrige Biegsamkeit / Stabilität	-	+	++	+++
4	Robust gegen kleine mechanische Schäden	-	-	++	++
1	Optische Ästhetik	-	-	++	++
37	Niedriges Gewicht	+++	++	+	-
					Δ Auswahl

Tabelle 4 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für die Abdeckungen. Es wird das Acrylglas XT 5 mm gewählt. Das Material hat eine gute optische Wirkung. Die Optik bleibt konsistent auch bei kleinen mechanischen Schäden. Beim Anschrauben von Geräten mit größerem Gewicht behält das Material seine Form. Der Hersteller kann eine Langzeitunterstützung anbieten, eine detaillierte Dokumentation ist vorhanden. Da diese Komponenten robust sein sollen, ist das höhere Gewicht der Platten zweitrangig. Acrylglas kann für die Anwendung leichter bearbeitet werden, da das Material weniger weich ist als MDF Holzplatten.

4.1.2 Gerüst

Das Gerüst bildet die haltende Struktur des Demonstrators. Für das Gerüstmaterial stehen zwei Materialvorschläge zur Verfügung:

- Aluminiumprofile: Diese Profile lassen sich durch Nutensteine und spezielle Führungen leicht zusammenschrauben.
- Eigenbau aus Metall- oder Holzstangen: Das Material wird auf arbiträre Weise bearbeitet und mit Schrauben und Muttern verbunden.

²² Mitteldichte Holzfasplatte

Tabelle 5: Vergleich von Materialansätzen für das Gerüst

Nr.	Anforderungen	Eigenbau aus Metall oder Holz	Aluprofile
27	Gut dokumentiert / es gibt Anwendungsbeispiele	--	++
28	Langzeitunterstützung des Herstellers / der Community	--	++
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	--	++
1	Optische Ästhetik	+	++
			Δ Auswahl

Tabelle 5 zeigt einen Vergleich zwischen den betrachteten Wahlmöglichkeiten für das Gerüstmaterial. Es wird der Ansatz der Aluprofile gewählt. Die Aluminiumprofile erlauben es auf eine einfache Weise Verbindungen herzustellen. Aluminiumprofile sind auf lange Sicht gut unterstützt und dokumentiert. Durch Standardisierung in der Bauweise und wenig Aufwand in der Entwicklung ist es viel leichter und schneller, ein optisch gut wirkendes Gerüst zu konstruieren.

Der gesamte Demonstrator soll aus ergonomischen Gründen in zwei Teile aufgeteilt werden. Es wird ein Unter- und ein Obergestell definiert. Das Obergestell soll die Präsentationselemente und die dafür benötigte elektronische Infrastruktur tragen, während das Untergestell als Stütze für das Obergestell dient.

Obergestell

Das Obergestell trägt die gesamte Elektronik und soll deswegen möglichst wenig Volumen einnehmen um mehr Elemente aufnehmen zu können. Das Material darf dennoch bei größerem Gewicht nicht durchbiegen. Wenn man die Grundfläche betrachtet, wird folgende Reihenfolge der Elemente innerhalb des Obergestells angenommen: Profil – Komponente – Profil. Da eine Komponente geringer als 150 mm tief sein soll, bleiben 50 mm für die Seitenlänge der Grundfläche von zwei Profilen. Es werden zwei Profile betrachtet: Profil 5 mit 20 mm x 20 mm Grundfläche und Profil 6 mit 30 mm x 30 mm Grundfläche.

Tabelle 6: Vergleich von Profilen für das Obergestell

Nr.	Anforderungen	Profil 5	Profil 6
37	Niedriges Gewicht	++	+
34	Kleine Maße	++	+
33	Seitenlänge der Grundfläche von zwei Profilen ist unter 50 mm	++	-
3	Niedrige Biegsamkeit / Stabilität	+	++
		Δ Auswahl	

Tabelle 6 zeigt einen Vergleich zwischen den betrachteten Wahlmöglichkeiten für die Obergestell-Profile. Es wird das Profil 5 gewählt. Die Seitenlänge der Grundfläche von zwei Profilen ist geringer als 50 mm. Dieses Profil ist biegsamer als Profil 6. Biegetests haben gezeigt, dass dieses Profil ausreicht um die Struktur des Obergestells konsistent halten zu können.

Untergestell

Das Untergestell soll als Stützung für das Obergestell dienen und muss massiver ausgelegt werden, um das Obergestell tragen zu können. Deshalb kann es auch größere Maße besitzen. Es

werden zwei Profile der schwereren Klasse betrachtet: Profil 8 mit 40 mm x 40 mm Grundfläche und Profil 10 mit 50 mm x 50 mm Grundfläche.

Tabelle 7: Vergleich von Profilen für das Untergestell

Nr.	Anforderungen	Profil 8	Profil 10
37	Niedriges Gewicht	++	+
3	Niedrige Biegsamkeit / Stabilität	++	+++
		Δ Auswahl	

Tabelle 7 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für die Untergestell-Profile. Der Spezifikation zur Folge hat Profil 8 eine ausreichende Stabilität für das Tragen von höheren Gewichten. Außerdem ist Profil 8 leichter als Profil 10 und damit vorteilhafter für das Gesamtgewicht des Demonstrators.

4.1.3 Halterungssystem

Bei dem Halterungssystem handelt es sich um einen Mechanismus für den standardisierten schraublosen Austausch von Komponenten im Inneren des Aufbaus. Es gibt zwei Ansätze:

- Eigenes Halterungssystem, das aus Platten mit speziell dafür konstruierten Halterungen besteht.
- Hutschiene mit der Profilgröße 35 mm x 7,5 mm.

Tabelle 8: Vergleich von Halterungssystemen

Nr.	Anforderung	Eigenes Halterungssystem	Hutschiene
27	Gut dokumentiert / es gibt Anwendungsbeispiele	--	++
28	Langzeitunterstützung des Herstellers / der Community	--	++
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	--	+
			Δ Auswahl

Tabelle 8 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für die Halterungssysteme am Beispiel der Hutschiene. Die Hutschiene erlaubt es, Geräte auf einfache Weise zu integrieren und auszutauschen. Für Komponenten, die über keinen Hutschieneanschluss verfügen, müssen dennoch spezielle Adapter erworben oder angefertigt werden. Der Hutschiene-Standard ist vom Hersteller und der Community gut dokumentiert und auf Langzeit unterstützt.

Die Hutschiene hat eine Nachfolgeanforderung: Alle Geräte im Demonstrator sollten über einen Hutschieneanschluss verfügen. Falls ein Gerät nicht vom Werk aus einen Hutschieneanschluss besitzt, sollte ein Adapter erworben oder entwickelt werden.

4.2 Kommunikationssystem des Anwendungsfalls

In diesem Unterkapitel werden Geräte und Software vorgestellt, die für die Kommunikation unter den Komponenten des Anwendungsfalls dienen sollen.

4.2.1 Middleware

Die Middleware stellt eine Abstrahierung der Kommunikation zwischen Avionics Computern dar. (vgl. 2.1.1). Es werden Nachrichtenpakete zum Informationsaustausch verwendet (vgl. 2.1.1). Weiterhin sollte es möglich sein, mit der Middleware innerhalb eines Computers einzelne Programmteile zu abstrahieren und zu verbinden (vgl. 2.1.1). Im Kontext des Demonstrators erlaubt die Middleware zusätzlich die einfache Integration von grafischen Frameworks. Hierfür gibt es zwei Ansätze:

- Robot Operating System (ROS)
- Message Queuing Telemetry Transport (MQTT)

ROS hat den Schwerpunkt in der roboter-technischen Entwicklung, es kann aber generell gut als Middleware eingesetzt werden (vgl. 2.1.3). MQTT ist ein Kommunikationsframework für IoT Umgebungen.

Tabelle 9: Vergleich von Middleware

Nr.	Anforderung	ROS	MQTT
27	Gut dokumentiert / es gibt Anwendungsbeispiele	++	++
28	Langzeitunterstützung des Herstellers / der Community	++	++
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+++	++
70	Repositorium an kompatiblen Softwarepaketen / Paketmanager	++	-
65	Abstrahierung der Kommunikation zwischen Avionics Computern	+++	++
66	Abstrahierung der Kommunikation innerhalb eines Avionics Computers	+++	++
67	Abstrahierung der Kommunikation zwischen einem Avionics Computer und einem Data Module	+++	++
68	Repositorium an Nachrichtentypen und Erstellung eigener Nachrichtentypen	++	++
69	Integration von grafischen Oberflächen	++	-
13	Werkzeuge für Entwicklung, Diagnose und Test	+++	+
16	Unterstützung von interpretierten Programmiersprachen	+++	+
		Δ Auswahl	

Tabelle 9 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für die Middleware. Es wird ROS gewählt. ROS hat ein großes Repositorium an Softwarepaketen, die man über einen Paketmanager installieren kann und es stehen die geforderten Kommunikationsabstrahierungen zur Verfügung. Es gibt die Möglichkeit, aus einem Repositorium von vordefinierten Nachrichtentypen zu wählen und eigene Nachrichtentypen zu erstellen. Zahlreiche Werkzeuge sind vorhanden, die bei der Entwicklung, der Diagnose und bei Tests unterstützen können. Außerdem unterstützt ROS zahlreiche Bindungen von interpretierten Programmiersprachen. ROS ist gut dokumentiert und verfügt über viele Anwendungsbeispiele. Eine Langzeitunterstützung des Herstellers und der Community ist vorhanden.

Durch die Entscheidung für ROS entsteht die Folgeanforderung für ein ROS unterstützendes Betriebssystem.

4.2.2 ARINC 429 Ersatz

Für die Kommunikation zwischen den Avionics Computern sollte ein Ersatz gefunden werden, der ARINC 429 nachbildet und gleichzeitig hardwarenah ist. Dafür gibt es zwei Möglichkeiten:

- Usbtin CAN-USB Adapter von Fishl: Dies ist eine Platine mit einem speziell angefertigten Chip. Dieser Chip kann über das CAN Protokoll mit anderen CAN Mitgliedern kommunizieren. Die Umsetzung der Adressierung und der Verbindung für ARINC 429 muss in SW erfolgen.
- UART-USB Adapter auf Basis von STM32: Dies ist eine Platine mit einem Mikrokontroller, der über zahlreiche UART Kanäle verfügt. Jeder UART Kanal dient als ein Verbindungskanal. Für die Kommunikation muss ein eigenes Protokoll und die Adressierungsfunktion implementiert werden.

Tabelle 10: Vergleich von ARINC 429 Ersatz

Nr.	Anforderung	Fishl Usbtin CAN-USB Adapter	STM32 UART-USB Adapter
27	Gut dokumentiert / es gibt Anwendungsbeispiele	+++	-
28	Langzeitunterstützung des Herstellers / der Community	+++	-
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	++	--
14	Hardwareabstrahierung und Treiber stehen zu Verfügung	+++	-
73	Hutschienenanschluss	-	-
		Δ Auswahl	

Tabelle 10 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für den ARINC 429 Ersatz; Es wird der CAN-USB Adapter gewählt. Er ist einfach einzusetzen, da nur wenige Kabel für die Verbindung der Kommunikationsmitglieder benötigt werden. Es gibt eine Hardwareabstrahierung und Treiber für verschiedene Betriebssysteme. Der Aufwand in der Entwicklung und Implementierung ist gering, da lediglich eine Abstrahierung der CAN Kanäle für die Adressierung und Verbindungsnachbildung von ARINC 429 durchgeführt werden muss. Im Vergleich mit dem CAN Adapter benötigt der STM32 UART-USB Adapter eine komplette Entwicklung der Firmware und Kommunikationsprotokolle vom Grund. Der CAN Adapter muss mit einem Hutschienenanschluss und entsprechendem Hutschienenadapter ausgestattet werden.

4.3 Komponenten des Anwendungsfalls

In diesem Unterkapitel werden Komponenten gewählt, die für die Nachbildung der Flugzeugreferenzarchitektur benötigt werden.

4.3.1 Sensoren

Statischer Luftdrucksensor und Gesamtlufttemperatursensor

Ziel ist der Ersatz des statischen Luftdrucksensors und des Gesamtluftdrucksensors durch funktionale Alternativen. Da die Gesamtlufttemperatur und der statische Luftdruck bei gleichen Raumbedingungen gemessen werden, lassen sich diese Werte mit einem kombinierten Sensor einfacher messen. Dafür gibt es zwei Varianten, nämlich den Adafruit BME280 und den Sparkfun

MS8607. Beide Geräte können Umgebungsluftdruck und Umgebungstemperatur messen. Beide werden über den I2C Kommunikationsbus angesprochen.

Tabelle 11: Vergleich der kombinierten Sensoren

Nr.	Anforderungen	Adafruit BME280	Sparkfun MS8607
27	Gut dokumentiert / es gibt Anwendungsbeispiele	+++	++
28	Langzeitunterstützung des Herstellers / der Community	+++	+++
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+++	+++
14	Hardwareabstrahierung und Treiber stehen zur Verfügung	+++	++
45	Messgenauigkeit des Gesamtlufttemperatursensors mindestens 0,5 Grad	++	-
47	Messgenauigkeit des statischen Luftdrucksensors mindestens 0,5 mbar	++	++
44	Messbereich des Gesamtlufttemperatursensors mindestens 0 – 50 Grad	++	++
46	Messbereich des statischen Luftdrucksensors mindestens 950 – 1060 mbar	++	++
		Δ Auswahl	

Tabelle 11 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für einen kombinierten Sensor. Es wird der Sensor von Adafruit gewählt. Im Vergleich mit dem Sparkfun Sensor erfüllt der Sensor von Adafruit die geforderte Genauigkeit für die Temperaturmessung. Er ist besser dokumentiert und hat mehr detaillierte Anwendungsbeispiele. Der Adafruit Sensor wird auch langfristig von dem Hersteller und der Community unterstützt. Die Verdrahtung und Integration in das Programm erfordert wenig Aufwand.

Der kombinierte Sensor von Adafruit wird über I2C angesprochen. Dieser muss von dem jeweiligen Data Modul für den Datenaustausch unterstützt werden.

Bremstemperatursensor

Der Bremstemperatursensor soll ersetzt werden. Der Sensor muss über einen metallischen Mantel für den direkten Kontakt mit der offenen Flamme eines Bunsenbrenners verfügen.

Dafür gibt es zwei Möglichkeiten:

- Ein PT100 mit einem metallischen Mantel und mit einem speziellen Akquirierungschip, den MAX31865. Dieser Chip ist speziell für den RTD Platinensensor ausgelegt. Er sendet die akquirierten Daten mittels SPI an einen Mikrokontroller.
- Ein PT100 mit einem metallischen Mantel und mit einer dafür ausgelegten Messbrücke. Die Messbrücke wird mit Strom versorgt und mit dem ADC eines Mikrokontrollers gemessen.

Tabelle 12: Vergleich der Bremstemperatursensoren

Nr.	Anforderung	PT100 + MAX31865	PT100 + eigene Messbrücke
27	Gut dokumentiert / es gibt Anwendungsbeispiele	+++	+
28	Langzeitunterstützung des Herstellers / der	+++	-

	Community		
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	++	-
14	Hardwareabstrahierung und Treiber stehen zur Verfügung	++	-
49	Messgenauigkeit des Bremstemperatursensors mindestens 1 Grad	++	++
48	Messbereich des Bremstemperatursensors mindestens 0 – 200 Grad	++	++
		Δ Auswahl	

Tabelle 12 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für den Ersatz des Bremstemperatursensors. Dafür wird der PT100 mit dem MAX31865 gewählt. Mit dieser Entscheidung lässt sich relativ einfach der PT100 Sensor einbinden und eine konsistente Datenübertragung an einen Mikrokontroller garantieren. Eine eigene Messbrücke wäre zwar eine einfachere Lösung, würde aber mehr Zeit für Entwicklung und Tests erfordern. Der MAX31865 verfügt über eine einfache Hardwareabstrahierung und ist gut dokumentiert; eine Langzeitunterstützung des Herstellers und der Community ist gewährleistet.

Der MAX31865 wird über SPI angesprochen. Dieser muss von dem jeweiligen Data Modul für den Datenaustausch unterstützt werden.

Raddrucksensor

Der Raddrucksensor soll nachgebildet werden. Dafür gibt es zwei Varianten:

- Honeywell ABPMANN004BGAA5. Dies ist ein einfacher miniaturisierter Drucksensor mit einem Messbereich von 0 - 4 bar, der einem linearen Spannungsausgang von 0 - 5 V entspricht.
- Omega PXM309-007GI, ein Industriedrucksensor mit einem Messbereich von 0 - 7 bar, der den Stromwerten von 4 – 20 mA entspricht.

Tabelle 13: Vergleich der Raddrucksensoren

Nr.	Anforderung	Honeywell ABPMANN004BG AA5	Omega PXM309-007GI
27	Gut dokumentiert / es gibt Anwendungsbeispiele	++	++
28	Langzeitunterstützung des Herstellers / der Community	++	++
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+	--
14	Hardwareabstrahierung und Treiber stehen zur Verfügung	+	--
51	Messgenauigkeit des Raddruckes mindestens 10 mbar	++	++
50	Messbereich des Raddruckes mindestens 0 – 4 bar	++	++
		Δ Auswahl	

Tabelle 13 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für den Ersatz der Raddrucksensoren. Die Entscheidung fällt für den Sensor von Honeywell. Da er einen 5 V Ausgang hat, kann er leicht mit einem 5 V Mikrokontroller verwendet werden. Im Vergleich zu dem Honeywell Sensor benötigt der Omega Sensor weitere Hardware, um die 4 bis 20 mA in

einen Spannungsbereich umzuwandeln. Der Sensor von Honeywell hat die Langzeitunterstützung des Herstellers und der Community und ist ausreichend gut dokumentiert.

Der Drucksensor von Honeywell gibt die Druckinformation als eine Gleichspannung aus. Das Data Module muss über einen Analog-Digital-Wandler für die Datenakquirierung verfügen.

4.3.2 Computer

Avionics Computer Ersatz

Die Avionics Computer sollen durch einen entwicklungsfreundlicheren Computer ersetzt werden. Dieser Computer muss mit einem Betriebssystem ausgestattet werden, das den Einsatz von interpretierten Sprachen ermöglicht. Die Performance des Computers sollte eine effiziente Entwicklung der Software unterstützen können. Dafür werden zwei Geräte aus unterschiedlichen Computerklassen miteinander verglichen:

- Raspberry-PI 3B+ aus der Einplatinen-Computerklasse. Die gesamte Logik ist auf einer Platine integriert. Es gibt 1 GB RAM und eine vier Kern ARM CPU, die mit 1,4 GHz getaktet wird. Die Wahl der Betriebssysteme ist auf die Linux Familie beschränkt. Der PI ist nicht sehr leistungstark, reicht aber für einige Computer Vision und Steuerungsaufgaben aus.
- Intel NUC8i5BEH aus der Mini-PC Klasse. Eine 2,3 GHz getaktete, vier Kern Intel i5 CPU ist auf der Platine integriert. Ein handelsüblicher PC Arbeitsspeicher und eine Festplatte werden extern angeschlossen. Es kann ein bis zu 32 GB großer Arbeitsspeicher verbaut werden. Dieser PC wird normalerweise für stark leistungsbezogene Anwendungen eingesetzt.

Tabelle 14: Vergleich für den Ersatz eines Avionics Computers

Nr.	Anforderungen	Raspberry PI 3B+	Intel NUC8i5BEH
27	Gut dokumentiert / es gibt Anwendungsbeispiele	+++	++
28	Langzeitunterstützung des Herstellers / der Community	+++	+++
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+++	+++
22	Niedrige Strombedarfe	++	--
37	Niedriges Gewicht	+++	-
34	Kleine Maße	+++	++
53	Ausreichende Performance beim Hochfahren, Kompilieren und im Normalbetrieb	+	+++
73	Hutschienenanschluss	-	-
		Δ Auswahl	

Tabelle 14 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für einen Ersatz von Avionics Computern. Die Wahl fällt auf den Raspberry PI 3B+. Der PI ist leicht, klein und benötigt wenig Strom. Er ist zwar nicht so schnell getaktet wie der NUC, dafür reicht seine Leistung für ein schnelles Hochfahren, Kompilieren und Ausführen der Software. Der PI 3B+ ist gut dokumentiert und hat viele Anwendungsbeispiele. Eine Langzeitunterstützung des Herstellers und der Community ist vorhanden. Für die Integration und Inbetriebnahme des gewählten Computers ist

wenig Aufwand erforderlich. Für den Einsatz des Raspberry PI 3B+ wird ein Hutschienenadapter benötigt.

Die Wahl des PI 3B+ hat zur Folge, dass man auf eine begrenzte Anzahl von Linux-Betriebssystemen beschränkt ist. Deshalb muss das Betriebssystem mit dem PI 3B+ kompatibel sein.

Hutschienenadapter für die Avionics Computer

Da in 4.1.3 die Hutschientechnologie gewählt worden ist, muss der Raspberry PI 3B+ auf einer Hutschiene befestigt werden können. Es werden zwei Ansätze betrachtet:

- Hutschienengehäuse für Raspberry PI 3B+ von Joy-iT.
- Spezielle Hutschienenhalterung - eine 3D gedruckte selbst konstruierte Halterung.

Tabelle 15: Vergleich von Hutschienenadaptern für die Avionics Computer

Nr.	Anforderungen	Joy-iT Hutschienengehäuse	Spezielle Hutschienenhalterung
27	Gut dokumentiert / es gibt Anwendungsbeispiele	+++	-
28	Langzeitunterstützung des Herstellers / der Community	+++	-
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+++	--
37	Niedriges Gewicht	+	++
34	Kleine Maße	+	++
		Δ Auswahl	

Tabelle 15 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für einen Hutschienenadapter. Die Entscheidung fällt für das Gehäuse von Joy-iT. Die spezielle Halterung kann zwar optimierter im Sinne des Gewichtes und der Maße konstruiert werden, benötigt dafür aber mehr Aufwand in der Entwicklung. Das Joy-iT Gehäuse ist gut dokumentiert und wird vom Hersteller langfristig unterstützt.

Betriebssystem für Avionics Computer

Das Betriebssystem wird für den Avionics Computer benötigt. Die Wahl ist auf PI 3B+ kompatible Linux-Betriebssysteme beschränkt. Es werden zwei oft in der PI Community eingesetzte Distributionen miteinander verglichen:

- Ubuntu Mate: Dies ist eine Ausführung des Ubuntu Betriebssystems mit dem Raspberry PI kompatiblen Kernel. Diese Modifikation des Ubuntu Betriebssystems ist speziell für die begrenzten Ressourcen des PI 3B+ konfiguriert worden.
- Raspbian: Ein speziell für PI entwickeltes Betriebssystem mit vielen vorinstallierten Programmen für den einfachen Entwicklungsstart.

Tabelle 16: Vergleich der Betriebssysteme

Nr.	Anforderung	Ubuntu Mate	Raspbian
27	Gut dokumentiert / es gibt Anwendungsbeispiele	+++	+++
28	Langzeitunterstützung des Herstellers / der Community	+++	+++

15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+++	+++
54	Repositorium an kompatiblen Softwarepaketen / Paketmanager	+++	+++
16	Unterstützung von interpretierten Programmiersprachen	+++	+++
55	Grafische Oberfläche	+++	+++
72	Kompatibilität mit PI 3B+	+++	+++
71	Kompatibilität mit ROS	+++	--
		Δ Auswahl	

Tabelle 16 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für das Betriebssystem. Es wird das System Ubuntu Mate gewählt. Ubuntu Mate wird langfristig vom Hersteller und der Community unterstützt und verfügt über eine gute Dokumentation und viele Anwendungsbeispiele. Man kann Ubuntu einfach auf einem PI3B+ installieren und die Programmierung mit einer Vielzahl von interpretierten Programmiersprachen durchführen. Außerdem gibt es ein großes Repositorium an Bibliotheken und Paketen, das mit einem Paketmanager installiert werden kann. Die grafische Oberfläche ist integriert und vorkompilierte ROS Distributionen sind vorhanden. Die Kompatibilität mit dem PI 3B+ ist gegeben.

Data Module Ersatz

Data Modules sollen durch einen Mikrokontroller ersetzt werden. Es werden zwei Varianten betrachtet:

- Nucleo STM32 L476LG. Eine Entwicklungsplatine auf Basis von ARM Cortex M4 L476RG, die mit 80 MHz getaktet wird. Sie besitzt eine hohe Anzahl von Kommunikationsgeräten wie z.B. Timer, Analog-Digital-Wandler, Digital-Analog-Wandler sowie andere moderne Mikrokontroller-Technologien.
- Arduino Mega 2560: Eine Entwicklungsplatine auf Basis von Atmel ATmega2560, die mit 16 MHz getaktet wird. Sie besitzt die grundlegenden Kommunikationsgeräte, Timer und Analog-Digital Wandler. Diese Platine wird gerne in der Hobbywelt verwendet.

Tabelle 17: Vergleich von Ersatzmöglichkeiten für Data Modules

Nr.	Anforderung	Nucleo STM32 L476RG	Arduino Mega 2560
27	Gut dokumentiert / es gibt Anwendungsbeispiele	+++	+++
28	Langzeitunterstützung des Herstellers / der Community	+++	+++
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	-	++
57	Unterstützung von UART, I2C, SPI und ADC	+++	+++
14	Hardwareabstrahierung und Treiber stehen zur Verfügung	+	+++
73	Hutschienenanschluss	-	-
			Δ Auswahl

Tabelle 17 visualisiert den Vergleich zwischen den betrachteten Ersatzmöglichkeiten für Data Module. Es wird der Arduino Mega gewählt, da er eine einfachere Hardwareabstrahierung besitzt, mehr Anwendungsbeispiele hat und über mehr Support auf Langzeit durch den Hersteller und die Community verfügt. Das Data Module besitzt keinen Hutschienenanschluss.

4.3.3 Displays

Lokales Display

Das lokale Display dient der lokalen Verifikation des Messwertes. Der in der Display Unit angezeigte Messwert wird mit dem Wert auf dem lokalen Display verglichen. Für das lokale Display stehen zwei Geräte zur Auswahl bereit:

- Ein Matrix OLED-Display von Grove, das über eine Auflösung von 128 x 64 und eine 0,96" Displaydiagonale verfügt.
- Ein Vier-Zeilen LCD-Display von DEBO, das 20 x 4 Zeichen auf einer 2,83" Displaydiagonale darstellen kann. Der LCD wird über den Chip HD44780 angesteuert.

Beide Geräte können nur zwei Farben abbilden.

Tabelle 18: Vergleich von lokalen Displays

Nr.	Anforderung	OLED-Display	Vier-Zeilen LCD-Display
27	Gut dokumentiert / es gibt Anwendungsbeispiele	+	+
28	Langzeitunterstützung des Herstellers / der Community	++	++
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	++	+
1	Optische Ästhetik	++	+
14	Hardwareabstrahierung und Treiber stehen zur Verfügung	++	++
		Δ Auswahl	

Tabelle 18 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für lokale Displays. Aus optischen Gründen fällt die Entscheidung für das Grove OLED-Display. Die vorhandene Hardwareabstrahierung erlaubt es, dass das Display ohne Aufwand in ein Programm eingebunden werden kann. Eine ausreichende Dokumentation ist vorhanden genauso wie auch die langfristige Unterstützung durch den Hersteller.

Das lokale Display von Grove wird über I2C angesprochen. Dieser muss von dem jeweiligen Data Module für den Datenaustausch unterstützt werden.

Display Unit

Für die Display Unit des Cockpits sollte ein Ersatz gefunden werden, um sie durch einen Computer mit einem grafischen Betriebssystem ansteuern zu können. Da für den Avionics Computer Raspberry PI 3 gewählt wurde, sollte das Display dazu kompatibel sein. Es werden zwei Geräte betrachtet:

- Das offizielle Raspberry PI Touch Display mit einer 7" Diagonale und 800 x 480 Pixel Auflösung. Die Kommunikation findet über den DSI-Anschluss statt. Mit speziellen Schraubanschlüssen lässt sich ein Raspberry PI direkt an die Platine von dem Displayboard anschließen.
- Das Touch Display von 52pi mit einer 7" Diagonale und 1024 x 600 Pixel Auflösung. Die Verbindung zum PI erfolgt über einen HDMI- und USB-Anschluss.

Tabelle 19: Vergleich von Display Units

Nr.	Anforderung	Raspberry Display	52pi Display
27	Gut dokumentiert / es gibt Anwendungsbeispiele	++	++
28	Langzeitunterstützung des Herstellers / der Community	++	++
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+++	+
59	Längste Seite ungefähr sieben Zoll	+++	+++
60	Hohe Auflösung, mindestens 1280x720	--	-
72	Kompatibilität zu PI 3B+	+++	++
37	Niedriges Gewicht	+	++
1	Optische Ästhetik	+++	+
		Δ Auswahl	

Tabelle 19 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für die Display Units. Es wird das offizielle Raspberry Display gewählt. Durch die Raspberry DSI Verbindung und spezielle Schraubanschlüsse kann es mit wenig Aufwand in Betrieb genommen werden. Das Display ist gut dokumentiert und hat die Langzeitunterstützung des Herstellers und der Community. Das Gewicht ist bei diesem Display höher als bei dem Display von 52pi, hat aber eine bessere Optik durch den integrierten Rahmen. Die längste Seite des Displays ist ungefähr sieben Zoll lang. Die Auflösung bei dem Raspberry Display ist niedriger als bei dem des 52pi, sie bietet aber trotzdem ausreichenden Spielraum für das Arbeiten mit dem Betriebssystem.

4.4 Stromversorgungssystem

Als Hauptquelle für die Stromversorgung dient der 230 VAC Wechselstromanschluss. Die Komponentenspannung soll im Bereich von 0 bis 24 VDC variieren. Wegen des Spannungsunterschiedes muss ein Wandler eingesetzt werden. Die gewandelte Spannung soll im System verteilt werden. Für die Wandlung und Verteilung stehen zwei Ansätze zu Verfügung:

- Die zentralisierte Wandlung: Es gibt einen Spannungswandler, der von 230 VAC auf 24 VDC transformiert. Die gewandelte Spannung wird über speziell verlegte Kabel an die Verbraucher verteilt.
- Die dezentralisierte Wandlung: Die 230 VAC Quellspannung wird durch eine Steckdosenleiste verteilt und lokal durch einen 230 VAC auf 24 VDC Spannungswandler transformiert.

Tabelle 20: Vergleich von Stromversorgungssystemansätzen

Nr.	Anforderungen	Zentralisierte Wandlung	Dezentralisierte Wandlung
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+	++
40	Spannung im Inneren des Demonstrators unter 60 VDC	++	--
34	Kleine Maße	++	--
37	Niedriges Gewicht	++	--
		Δ Auswahl	

Tabelle 20 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für den Wandlungsansatz. Die Entscheidung fällt für den Ansatz der zentralisierten Wandlung, weil das Gewicht niedrig ist und die benötigten Maße klein gehalten werden können. Die zentralisierte Wandlung ist elektrisch sicherer, da 230 VAC nur an einer isolierten Stelle geleitet wird. Der Rest des Systems wird mit 24 VDC versorgt, also weniger als 60 VDC. Der zentralisierte Wandleransatz benötigt allerdings mehr Aufwand in der Realisierung.

4.4.1 Anschluss

Das Ziel ist, eine Kaltgerätebuchse mit einem Hauptstromschalter und einer Sicherung in den Demonstrator zu integrieren. Hierfür bieten sich zwei Ansätze an:

- Ein kombinierter Anschluss von QLOUNI: Kaltgerätebuchse, Schalter und Sicherungseinschub sind in einem Gehäuse integriert.
- Separate Bauteile, Kaltgerätebuchse, Schalter und Sicherungseinschub werden einzeln in das Gehäuse des Demonstrators integriert.

Tabelle 21: Vergleich von Ansätzen für den Wechselstromanschluss

Nr.	Anforderungen	Kombinierter Anschluss	Separate Geräte
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	++	-
34	Kleine Maße	++	++
		Δ Auswahl	

Tabelle 21 veranschaulicht den Vergleich zwischen den betrachteten Wahlmöglichkeiten für den Wechselstromanschluss. Es wird der kombinierte Anschluss gewählt, weil dieser durch das einzelne Gehäuse viel leichter in den Demonstrator integriert werden kann.

4.4.2 Wandlung

Hauptnetzteil

Das Hauptnetzteil stellt für die elektronischen Komponenten des Demonstrators Strom zur Verfügung. Die Stromgrenze sollte entsprechend der Strombedarfe der Avionics Computer und der angeschlossenen Peripherie ausgelegt werden. Strommessungen bei einem Test haben gezeigt, dass für die gewünschte Anwendung bis zu 500 mA pro Raspberry PI 3B+ benötigt werden. Dafür wird ein CAN-USB Adapter und ein PI Touch Display angeschlossen und der Strom für die drei Betriebszustände Hochfahren, Kompilieren und Normalbetrieb gemessen. Der Strombedarf für 15 Computer beträgt 7,5 A bei Berücksichtigung eines Puffers von zusätzlich 2,5 A. Das Netzteil sollte also bis zu 10 A Strom zur Verfügung stellen können, um auch für zukünftige Anwendungen ausgerüstet zu sein. Es werden zwei hochwertige Hutschienengeräte mit 24 VDC Nennausgangsspannung, 10 A Grenzstrom und 240 W Leistung verglichen:

- Das Schaltnetzteil Uno Power PS von Phoenix Contacts.
- Das Schaltnetzteil DRP-240-24 von Mean Well.

Tabelle 22: Vergleich von Hauptnetzteilen

Nr.	Anforderungen	Uno Power	Mean Well
27	Gut dokumentiert / es gibt Anwendungsbeispiele	+++	+++
28	Langzeitunterstützung des Herstellers / der Community	+++	+++

15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+++	+++
37	Niedriges Gewicht	++	+
34	Kleine Maße	++	+
		Δ Auswahl	

Tabelle 22 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für das Hauptnetzteil. Es wird das Netzteil von Phoenix Contact gewählt. Obwohl beide Netzteile über vergleichbare Eigenschaften verfügen, ist das Netzteil von Phoenix Contact leichter und kompakter.

Zusätzlich sollen Schmelzsicherungen eingebaut werden um Kabel und Geräte zu schützen, sowohl auf der Primär- als auch auf der Sekundärseite des Netzteils. Das Netzteil hat einen Übersetzungsfaktor von 10. Strom von 10 A auf der Sekundärseite benötigt Strom von 1 A auf der Primärseite. Folgerichtig schützt man die Sekundärseite mit einer 10 A Sicherung, die Primärseite mit einer 1 A Sicherung.

Lokales Netzteil

Das lokale Netzteil hat die Aufgabe, die 24 VDC Spannung in eine niedrigere Gleichspannung umzuwandeln. In diesem Projekt wird die Gleichspannung für die Versorgung des Raspberry PI 3B+ benötigt. Der PI 3B+ hat einen Grenzspannungsbereich von 4,63 V bis 6 V. Bei 4,63 V wird die Unterspannung auf dem Display angezeigt. Bei 6 V wird die integrierte Überstromsicherung ausgelöst. Der Normalbetrieb bewegt sich innerhalb des Spannungsbereiches +/- 5 % der Nennspannung, was 4,75 V bis 5,25 V entspricht. Eine minimale Überschreitung dieser Werte ist tolerierbar. Daher sollte die Ausgangsspannung des lokalen Netzteils eine Stromwelligkeit von 50 mVpp haben. Es werden zwei DC-DC Spannungswandler miteinander verglichen.

- Pololu, Modell S18V20ALV, mit dem TPS55340 Chip.
- Modell LM2596S.

Tabelle 23: Vergleich von lokalen Netzteilen

Nr.	Anforderung	Pololu S18V20ALV (TPS55340 Chip)	LM2596S
27	Gut dokumentiert / es gibt Anwendungsbeispiele	++	++
28	Langzeitunterstützung des Herstellers / der Community	++	++
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	++	-
21	Stromwelligkeit unter 50 mVpp bei 5 V Nennausgangsspannung	++	-
73	Hutschiennenanschluss	-	-
		Δ Auswahl	

Tabelle 23 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für lokale Netzteile. Es wird das lokale Netzteil von Pololu gewählt. Der Wandler von Pololu erfüllt die geforderte Stromwelligkeit besser als der LM2596S, weil er einfacher zu verdrahten ist und Schraubanschlüsse zur Verfügung stehen. Das Netzteil kann problemlos erworben werden und ist ausreichend dokumentiert. Das Netzteil von Pololu verfügt über keinen Hutschiennenanschluss.

4.4.3 Verteilung

Die Verteilung ermöglicht es aus einer Quelle mehrere elektronische Komponenten mit Spannung zu versorgen. In diesem Fall wirken zwei Verteilungskategorien zusammen, nämlich die lokale und die globale Spannungsverteilung. Die globale Verteilung liefert Strom an die Anwendungsmodule. Die lokale Verteilung versorgt Komponenten mit Strom innerhalb eines Anwendungsmoduls. Diese Aufteilung erlaubt es, die Übersicht zu behalten und eine einfache standardisierte Erweiterbarkeit zu ermöglichen.

Globale Spannungsverteilung

Unter Punkt 4.4 wurde die Wahl für den zentralisierten Wandlungsansatz begründet. Die gewandelte Spannung soll über eine Kabelstruktur im Demonstrator verteilt werden. In diesem Fall gibt es drei Netzformen für die Verteilung:

- Parallel: Von der Quelle zum Verbraucher wird sternförmig jeweils ein Kabel gelegt.
- Bus: Eine Leitung wird durch den gesamten Demonstrator verlegt und an lokalen Stellen abgezweigt.
- Beides: Eine Kombination aus Parallel- und Busschaltung.

Tabelle 24: Vergleich von Verteilungsarten

Nr.	Anforderung	Parallel	Bus	Kombination von Parallel- und Busschaltung
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+++	+	++
37	Niedriges Gewicht	--	++	+
34	Kleine Maße	-	++	+
24	Niedrige Auslastung durch den Strom	++	--	+
				Δ Auswahl

Tabelle 24 zeigt uns den Vergleich zwischen den betrachteten Wahlmöglichkeiten für die Verteilungsarten. Es wird der kombinierte Ansatz aus Bus- und Parallelschaltung, also ein Kompromiss, gewählt. Diese Kombination verfügt über weniger Gewicht und kleinere Maße als die reine Parallelschaltung. Sie hat eine geringere Auslastung durch hohe Ströme als bei einer reinen Busschaltung. Dadurch kann Gewicht gespart und mehr Strom geleitet werden ohne einen Umbau durchführen zu müssen.

Zusätzlich soll ein Kabelquerschnitt anhand der Stromgrenze von 10 A gewählt werden. Laut VDE 0100 Teil 430 ist bei beweglichen Mehraderleitungen die Kabelgruppe 2 zu berücksichtigen. Bei einem Strom von 10 A in einer Kupferleitung sollte mindestens ein Nennquerschnitt von 1 mm² gewählt werden.

Lokale Spannungsverteilung

Die lokale Spannungsverteilung ist die Fortsetzung der globalen Verteilung. Es sollen innerhalb einer lokalen elektronischen Gruppe auf einfache Weise Spannungsanschlüsse an die Endverbraucher verteilt werden. Es werden zwei Ansätze betrachtet:

- Die Reihenklemme: Standardisierte Einheiten können auf eine Hutschiene aufgesetzt werden, die Kabel lassen sich mit Klemm- oder Schraubanschlüssen leicht verbinden. Man kann die Anschlüsse beliebig konfigurieren.
- Die „spezielle“ Verbindungsplatine: Eine Lochrasterplatine mit Steckern und Buchsen, die parallel geschaltet sind.

Tabelle 25: Vergleich von Ansätzen für die lokale Spannungsverteilung

Nr.	Anforderung	Reihenklamme	Spezielle Verbindungsplatine
15	Wenig Aufwand in der Entwicklung, Implementierung und Anwendung	+++	--
37	Niedriges Gewicht	+	++
34	Kleine Maße	+	++
73	Hutschienenanschluss	+++	-
			Δ Auswahl

Tabelle 25 zeigt den Vergleich zwischen den betrachteten Wahlmöglichkeiten für die lokale Spannungsverteilung. Es wird die Reihenklamme gewählt. Dieser Ansatz benötigt mehr Bauraum und Gewicht als eine eigenkonstruierte Platine. Dafür erfordert sie weniger Aufwand bei der Anwendung, da Verbindungen durch ein einfaches Klemmen ohne Lötarbeiten eingerichtet werden können. Der Hutschienenanschluss erlaubt auch eine einfache mechanische Integration in den Demonstrator. Er verfügt über eine ausreichende Dokumentation und eine gute Langzeitunterstützung durch den Hersteller.

5 Entwicklung

Das vorliegende Kapitel beschäftigt sich mit der Implementierung und richtet sich nach dem grundlegenden Strukturkonzept, das in der Konzeption (Abbildung 20) ausgearbeitet wurde. In Abschnitt 5.1 wird der Demonstrator mit seinem mechanischen Trägersystem und dem Stromversorgungssystem ausgearbeitet. Abschnitt 5.2 behandelt die Entwicklung des Beispielfalls und geht unter anderem auf die Verbindung zwischen Computern, Sensoren und Displays ein.

Die Umsetzung dieser Kategorien passiert auf der mechanischen, elektronischen und der programm-technischen Ebene. Wenn möglich, wird zuerst ein Modell oder ein Schaltplan erstellt und die jeweilige Komponente erst auf dessen Basis konkret umgesetzt.

5.1 Demonstrator

Für die Umsetzung des Demonstrators wird als erstes der mechanische Aufbau entworfen und gefertigt. Auf der Grundlage so entstandenen Gerüsts wird dann die Stromversorgung geplant, eingebaut und verdrahtet. Das mechanische Trägersystem beschreibt die Konstruktion und Umsetzung von dem Gerüst und den Abdeckungen. Die Stromversorgung behandelt die Umwandlung und Verteilung von Spannung.

5.1.1 Mechanisches Trägersystem

Das mechanische Trägersystem hat zwei Funktionen: Es soll einerseits alle relevanten Komponenten tragen, andererseits präsentabel wirken. Es werden Aluminiumprofile für das Gerüst eingesetzt und Acrylplatten für die Abdeckungen.

Aufteilung in logische Einheiten

Für die Organisation innerhalb des Aufbaus soll der Demonstrator auf zwei Arten logisch aufgeteilt werden:

- Aufteilung in Funktionsebenen
- Aufteilung in Anwendungsmodule

Die Funktionsebenen sollen den mechanischen Aufbau in der Tiefe aufteilen. Es gibt insgesamt drei Funktionsebenen:

- Präsentationsebene
- Prozessebene
- Serviceebene

Die Präsentationsebene nimmt alle visuellen Komponenten auf, die für die Vorführung des Forschungsergebnisses zuständig sind. Die Prozessebene beinhaltet alle Verarbeitungs- und Versorgungskomponenten, wie Computer und Netzteile. Die Serviceebene dient dem Zugang zu dem Inneren des Demonstrators und kann weitere Komponenten aufnehmen, die nicht für die Präsentation zuständig sind.

Die Anwendungsmodule sollen die Anwendung modularisieren. Jedes Anwendungsmodul gruppiert alle Komponenten, die logisch miteinander zusammenhängen. Dabei können die Komponenten elektronischer, mechanischer und programm-technischer Natur sein. Die Komponenten können auf allen drei Funktionsebenen vorkommen. Durch diese Aufteilung

können unterschiedliche Systeme separat betrachtet werden. Nach dem Beenden der Forschungsarbeit kann das jeweilige Anwendungsmodul mit einem anderen Inhalt ersetzt werden. Das Obergestell wird in neun Anwendungsmodule in einem 3 x 3 Schachbrettmuster aufgeteilt. So wird einerseits ausreichend Fläche für eine Anwendung gegeben, andererseits kann eine ausreichende Modularisierung stattfinden.

CAD Modell

Das CAD Modell soll in erster Linie helfen, die vorgesehene Wirkung der Optik einzuschätzen. Es ist auch ein wichtiger Schritt vor der Tätigung der Bestellung, da exakte Längen und Profilgrößen angegeben werden müssen.

Es sollen das Ober- und das Untergestell konstruiert werden. Der gesamte Aufbau soll höchstens 2 m x 2 m x 0,2 m groß sein, wobei die Höhe an die Möglichkeit des Tragens durch einen Türrahmen angepasst werden soll. Durch einen Ergonomietest wurde festgestellt, dass ein Puffer von ca. 100 mm ausreicht. Der Demonstrator kann durch eine Tür getragen werden, und die vorgegebene Höhe wird dabei minimal verändert. Die neue Höhe soll somit ca. 1900 mm betragen. Der Demonstrator wird in der Höhe in zwei Hälften geteilt: Die untere Hälfte stellt das Untergestell, die obere Hälfte das Obergestell dar. Es werden für das Obergestell etwa 2 m x 1 m Maße angenommen. Für die Verbesserung der optischen Wirkung wird das Obergestell jedoch an das Seitenverhältnis der 16:9 Bildschirme angelehnt. Somit hat die Präsentationsfläche des Obergestells eine Breite von 1920 mm und eine Höhe von 1080 mm.

Für das Obergestell werden zwei Typen des Profils 5 verwendet nämlich mit 20 mm x 20 mm und 40 mm x 20 mm Grundfläche. Das breitere Profil wird benötigt, um zwei Vorderabdeckungen bündig aneinander zu reihen. Für das Untergestell wird das Profil 8 verwendet mit 80 mm x 40 mm Grundfläche. Zusätzlich werden entsprechende Aluminiumwinkel für das Zusammenfügen von Profilen eingesetzt.

Das Obergestell des Demonstrators soll flächendeckend mit Acrylplatten abgeschlossen werden. Die Seiten, das Dach und die Serviceebene werden zwischen zwei Profilen verbaut. Die Abdeckungen der Präsentationsebene sollen bündig aneinandergereiht werden. Die untere Seite des Obergestells wird zwecks Lüftung der Geräte nicht abgeschlossen.

In Abbildung 21 ist das Ergebnis der CAD-Konstruktion zu sehen – mit allen genannten Maßen und Aufteilungen. Es wird die Vorder- und die Seitensicht gezeigt. Zwei Dinge können hier betrachtet werden. Einerseits die Gesamtmaße, andererseits die logischen Aufteilungen. Das ist die Aufteilung in Ebenen und Aufteilung in Anwendungsmodule.

Umsetzung

Die Profile werden bestellt und gemäß dem CAD Modell zusammengeschraubt. Da sich die Position der Profile an einigen Stellen minimal verschieben könnte, werden die Schrauben schrittweise angezogen. Zuerst wird das Untergestell zusammengeschraubt, dann das Obergestell und schließlich werden die beiden Hälften miteinander verbunden.

Die Hutschiene wird an die Breite eines Anwendungsmoduls angepasst und mit einem Profil verstärkt, damit die aufgesetzten Geräte stabil getragen werden können. Die Hutschiene wird auf der Prozessebene aufgesetzt. Es werden jeweils zwei Hutschienen innerhalb eines Anwendungsmoduls verbaut.

Für die vorderen Abdeckungen werden die Acrylglasplatten überdimensioniert bestellt und mit einem CO₂ Industrielasercutter geschnitten. Durch die Überdimensionierung können die Bohrungen passgenau gelasert werden. Die Seiten-, Dach- und Serviceabdeckungen können nicht in den Lasercutter eingelegt werden, da nicht ausreichend Arbeitsraum zur Verfügung steht. Hierfür werden spezielle Schablonen aus Holz erstellt, um die Positionen der Bohrungen dann manuell übertragen zu können. Diese werden so dimensioniert, dass sie passgenau an die Kanten der Acrylplatten angelegt werden können. So kann eine richtige Referenzierung durchgeführt werden.

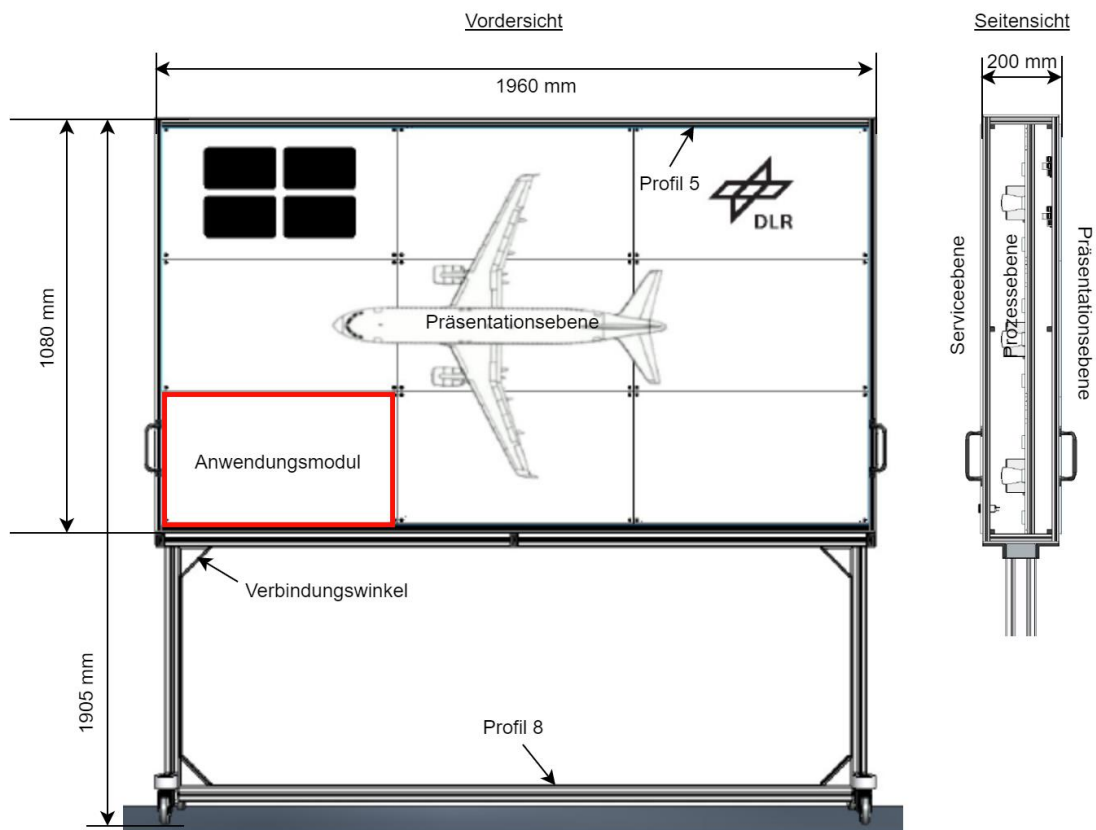


Abbildung 21: CAD-Konstruktion des mechanischen Aufbaus

5.1.2 Stromversorgungssystem

Bei dem Stromversorgungssystem gibt es zwei Stromverteilungsmaßstäbe: Die globale und die lokale Stromversorgung. Die globale Stromversorgung liefert Energie an die einzelnen Anwendungsmodule. Die lokale Stromversorgung wirkt innerhalb eines Anwendungsmoduls.

Globale Stromversorgung

In Abbildung 22 ist der Schaltplan für die globale Stromversorgung zu sehen. Die Kaltgerätebuchse stellt 230 VAC zur Verfügung. Der Hauptschalter S1 aktiviert die Wechselspannung. Die Sicherung F1 schützt die Kabel auf der Primärseite. Das Hauptnetzteil richtet die Wechselspannung und transformiert sie zu einer niedrigeren Gleichspannung von 24 VDC. Die Sicherung F2 schützt die Kabel auf der Sekundärseite. Die globale Spannung von 24 VDC wird zuerst an die drei Versorgungsstränge verteilt und dann jeweils an drei Anwendungsmodule geleitet, so dass alle neun Anwendungsmodule mit Spannung versorgt werden.

Die Komponenten werden entsprechend Abbildung 22 verdrahtet. Alle Verbindungsstellen haben entweder Klemm- oder Schraubverbindungen. Es ist sinnvoll, Aderendhülsen an den Enden der Kabel zu verwenden.

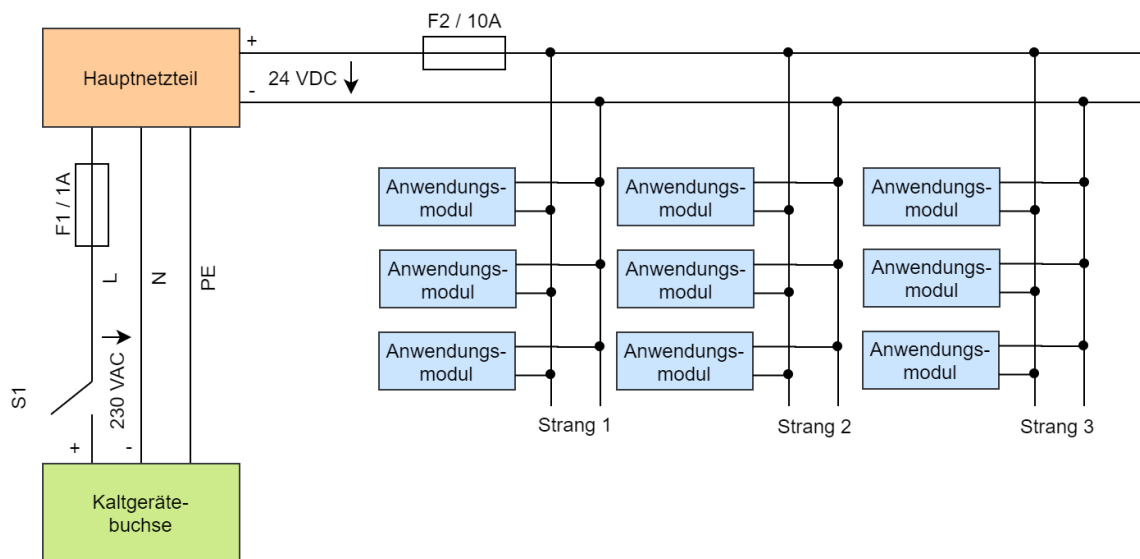


Abbildung 22: Schaltplan für die globale Stromversorgung

Zuerst wird die Kaltgerätebuchse in einer Seitenabdeckung eingebaut. Hierfür wird eine Holzschablone für die Positionierung benötigt, da wegen der Größe der Platte die Seitenabdeckung nicht in den CO2 Lasercutter eingelegt werden kann.

Dann wird das Hauptnetzteil auf die Hutschiene aufgesetzt. Die Richtlinien von DLR erfordern, dass ausschließlich eine Elektrofachkraft an der 230 VAC Wechselspannung arbeiten darf. Ein hausinterner Elektriker stellt die Verbindung zwischen der Kaltgerätebuchse und der Primärseite des Hauptnetztes her. Dabei wird in die Kaltgerätebuchse die Schmelzsicherung F1 mit 1A eingelegt. Der PE Schutzleiter wird mit allen leitenden Metallteilen des Gerüsts verbunden.

Die Sekundärseite des Hauptnetztes wird mit der Sicherung F2 über einen speziellen Adapter verbunden. In den Sicherungsadapter F2 wird eine 10 A Schmelzsicherung eingelegt. Der Sicherungsadapter und der negative Pol der Sekundärseite werden an eine globale Reihenklemme mit 10 parallel geschalteten Verbindungen angeschlossen.

Die Kabel werden zuerst, wie in Abbildung 22 dargestellt, in drei Stränge geteilt und entlang von drei Anwendungsmodulen durch den Kabelkanal durchgeführt. Innerhalb eines Anwendungsmoduls wird eine Lüsterklemme auf einem 3D gedruckten Adapter angeschraubt. Durch die Lüsterklemme erhöht sich die Übersichtlichkeit und Organisation. Durch sie wird auch die Spannung innerhalb eines Spannungsstranges an das nächste Anwendungsmodul weitergereicht.

In Abbildung 23 ist das Ergebnis der Verdrahtung zu sehen. Links ist die Verdrahtung der globalen Stromversorgung, rechts der Wechselstromanschluss abgebildet.

Lokale Stromversorgung

In Abbildung 24 sind zwei Verwendungsfälle innerhalb eines Anwendungsmoduls zu sehen. Die globale 24 VDC Spannung wird entweder direkt an die Verbraucher weitergeleitet oder zuerst

über einen DC-DC Wandler zu einer niedrigeren Gleichspannung transformiert und dann weitergegeben.

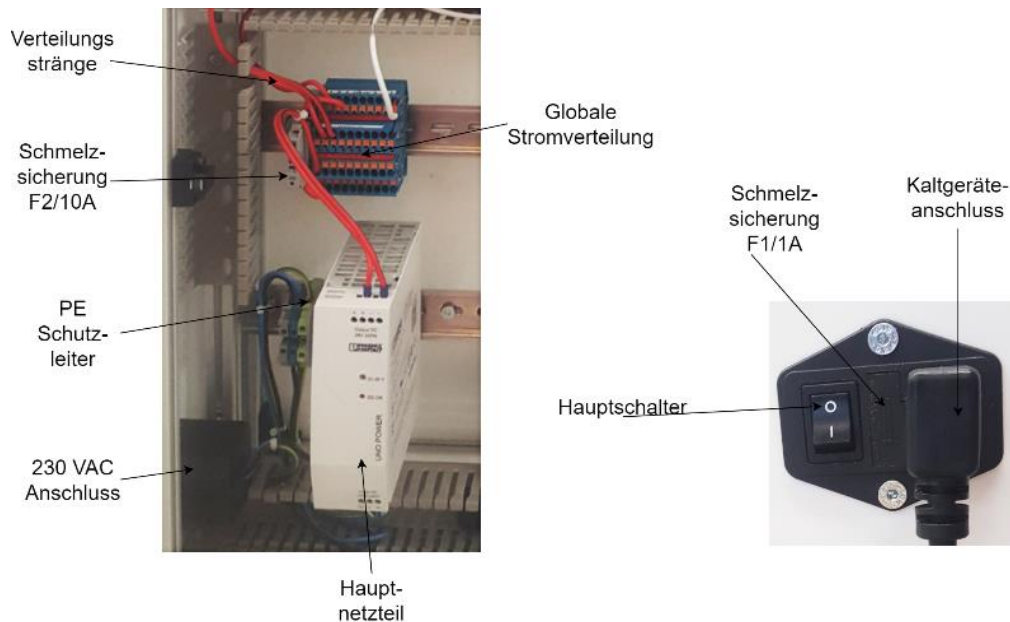


Abbildung 23: Die Umsetzung der globalen Stromverteilung

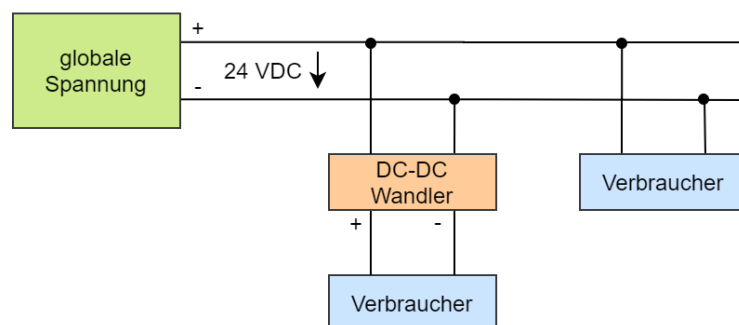


Abbildung 24: Schaltplan für die lokale Stromversorgung

Um die Verbindungsstruktur in Abbildung 24 umsetzen zu können, wird innerhalb eines Anwendungsmoduls eine Reihenklemme mit zehn parallel geschalteten Verbindungen auf die Hutschiene aufgesetzt. Die Reihenklemmen der Anwendungsmodule werden mit den entsprechenden Lüsterklemmen verbunden. Die konkrete Umsetzung findet im Unterkapitel 5.2 statt.

5.2 Die Beispielanwendung

In Abbildung 25 ist die schematische Übersicht über die Systeme der Beispielanwendung zu sehen. Das System ist in zwei Kategorien unterteilt. Das sind die Akquirierungseinheiten und ein verteiltes System von Avionics Computern. Die Akquirierungseinheit beinhaltet Sensoren, lokale Displays und Data Modules. Das verteilte System beinhaltet alle Avionics Computer, das Display und die CAN-Bus Adapter. Jede Akquirierungseinheit gehört zu einem entsprechenden Avionics Computer.

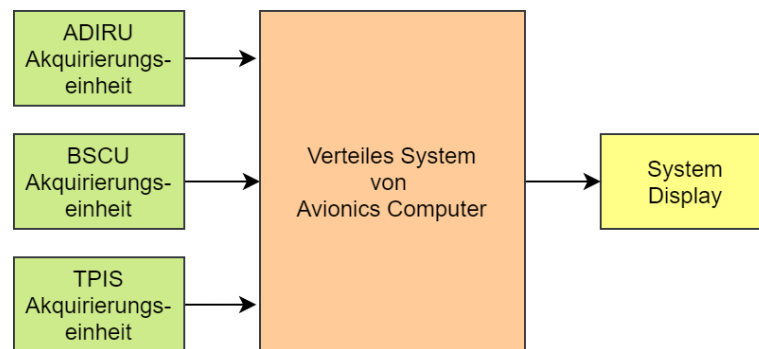


Abbildung 25: Übersicht über die Systeme der Beispielanwendung

5.2.1 Akquirierungseinheiten

Die Akquirierungseinheiten sollen an die jeweiligen Avionics Computer über USB angeschlossen werden. Die Kommunikation zwischen einem Data Module und dem Avionics Computer findet über das Programm „Rosserial“ statt. Dafür wird auf dem Avionics Computer ein serielles Gate in der ROS Umgebung gestartet, welches ROS Nachrichten über die serielle Schnittstelle sendet und empfängt.

Die Sensoren und lokalen Displays der jeweiligen Einheit sollen in einer visuellen Einheit zur besseren Optik und Handhabung beim Einbau zusammengefasst werden. Für die visuelle Einheit wird ein entsprechender Ausschnitt in der Platte eines Anwendungsmoduls gemacht. Die visuelle Einheit wird dann zusammen mit den Sensoren und Displays aufgeklebt.

ADIRU Akquirierungseinheit

Die ADIRU Akquirierungseinheit soll statische Druck- und Gesamtlufttemperaturwerte aufnehmen, auf einem lokalen Display anzeigen und an den ADIRU Avionics Computer per USB senden. Der Sensor und das lokale Display der ADIRU werden in einer visuellen Einheit integriert (Abbildung 26). Ein mit „Air Data“ gekennzeichnetes Gerät stellt den BME280 Kombisensor dar, das OLED-Display in der Mitte zeigt die aktuellen Messwerte. Für die ADIRU Akquirierungseinheit wird ein Anwendungsmodul der Präsentationsebene belegt. Für die visuelle Einheit wird dann ein entsprechender Ausschnitt in dem ADIRU Anwendungsmodul vorgenommen. Die visuelle Einheit wird dann reingelegt.



Abbildung 26: Visuelle Einheit des ADIRU Systems

In Abbildung 27 ist das Blockschaltbild der ADIRU Akquirierungseinheit zu sehen. Sie besteht aus dem kombinierten BME280 Sensor, einem Grove OLED Display und einem Arduino Mega. Der BME280 wird über I2C an den Arduino Mega angeschlossen und sendet ihm den aktuellen Umgebungsdruck und die Umgebungstemperatur. Ein OLED Display ist über denselben I2C Bus angeschlossen. Dieser bekommt von dem Arduino Mega aktuelle Sensorwerte der Temperatur und des Drucks für die Darstellung. Der Arduino sendet die gemessenen Werte über die serielle Schnittstelle mittels USB Verbindung an den ADIRU Raspberry PI. Für das Senden werden zwei ROS-Publisher erstellt, jeweils einer für Temperatur und Druck.

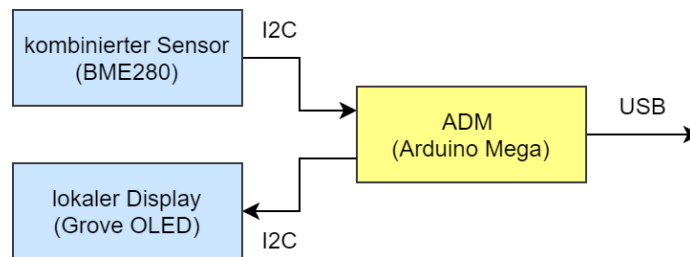


Abbildung 27: Verbindungsstruktur der ADIRU Akquirierungseinheit

In Abbildung 28 ist das Aktivitätsdiagramm von dem ADM zu sehen. Nach dem Mikrokontrollerstart werden der Sensor, die Displays und interne Zustände initialisiert und die ROS Topics deklariert. In der Schleife werden als erstes die Temperatur- und Druck-Sensorwerte akquiriert. Die Werte werden dann am lokalen Display angezeigt. Weiter werden sie an zwei Topics gesendet, den „/total_air_temperature_orig“ und „/static_air_pressure_orig“ Topic. Das „orig“ steht für „original“, was darauf hindeutet, dass der gesendete Wert nicht über ein Label verfügt. Dieser wird später in der ADIRU beigefügt. Anschließend wird ein Update der ROS-internen Zustände durchgeführt, 10 ms gewartet und der Neustart der Schleife initiiert.

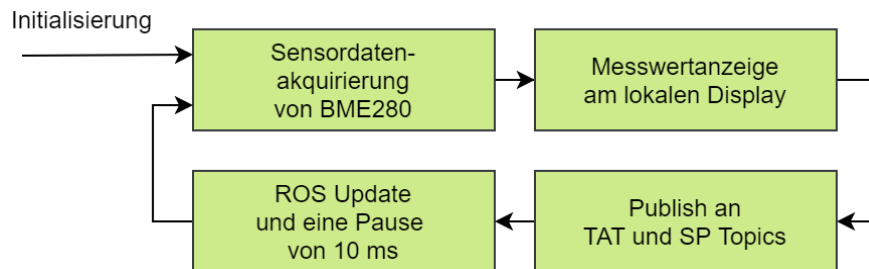


Abbildung 28: Aktivitätsdiagramm des ADM

BSCU und TPIS Akquirierungseinheiten

Die BSCU Akquirierungseinheit soll Bremstemperaturwerte aufnehmen, auf einem lokalen Display anzeigen und an den BSCU Avionics Computer per USB senden. Die TPIS Akquirierungseinheit soll Raddruckwerte aufnehmen, auf einem lokalen Display anzeigen und an den TPIS Avionics Computer senden. Für die BSCU und TPIS Akquirierungseinheiten wird ein Anwendungsmodul der Präsentationsebene belegt.

Die TPIS und die BSCU Akquirierungseinheit werden in einer visuellen Einheit zusammengefasst (Abbildung 29). Dabei werden die beiden Sensorwerte auf einem lokalen Display angezeigt. Die visuelle Einheit besteht aus einem PT100 (rechts), einem Honeywell Drucksensor (links) und einem lokalen Display (Mitte). Für die visuelle Einheit wird ein entsprechender Ausschnitt in der

Platte des BSCU/TPIS Anwendungsmoduls gemacht. Die visuelle Einheit wird dann zusammen mit den Sensoren und Displays aufgeklebt.

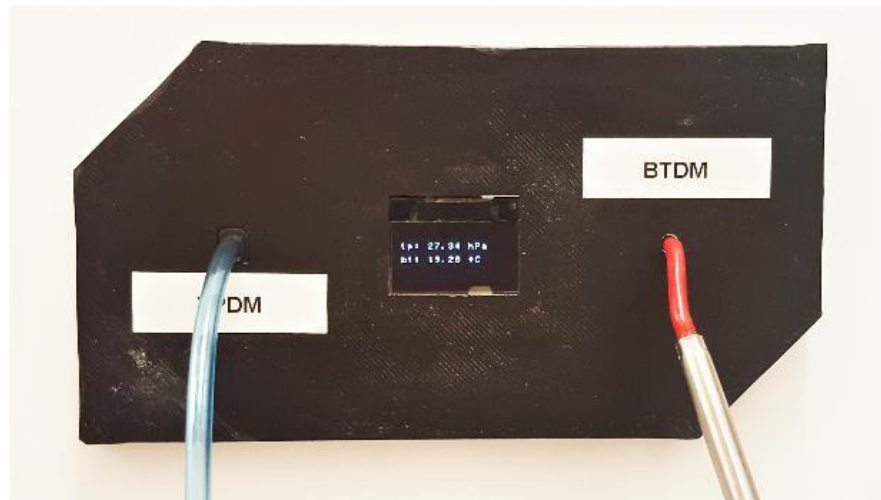


Abbildung 29: Visuelle Einheit des BSCU/TPIS Systems

In Abbildung 30 ist das Blockschaltbild der BSCU und der TPIS Akquirierungseinheiten zu sehen. Der PT100 Sensor wird über die Dreileitertechnik an den MAX31865 angeschlossen. Der MAX31865 wird so konfiguriert, dass er einen Dreileitersensor verarbeiten kann. Er wird über SPI an den BTDM Arduino Mega angeschlossen. Der analoge Ausgang des Honeywell Drucksensors wird an den Analog-Digital-Wandler des TPDM Arduino Mega angeschlossen. Das OLED Display wird über I2C an den TPDM Arduino Mega angeschlossen. Zwischen dem TPDM und BTDM wird eine weitere I2C Verbindung aufgebaut, um den Messwert von dem BTDM an den TPDM zu senden. Dies dient der weiteren Anzeige auf dem lokalen Display. Beide Arduino Mega werden mit den entsprechenden Avionics Computern über die serielle Schnittstelle mittels USB verbunden. Der BTDM wird mit BSCU Avionics Computern verbunden und der TPDM mit TPIS Avionics Computern.

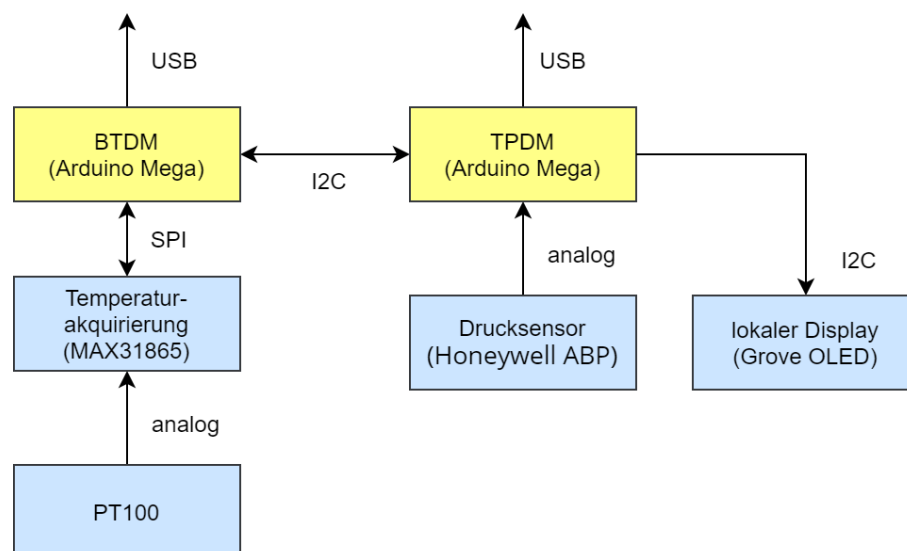


Abbildung 30: Verbindungsstruktur der BSCU und TPIS Akquirierungseinheit

In Abbildung 31 ist das Aktivitätsdiagramm des TPDM zu sehen. Nach dem Mikrokontrollerstart werden das Display, interne Zustände und die I2C Verbindung zum BTDM initialisiert. Zusätzlich

wird der ROS Topic für Raddruck deklariert. Innerhalb der Schleife werden zuerst die Sensordaten durch den Analog-Digital-Wandler akquiriert. Dann wird der digitale Spannungswert entsprechend der Formel, die vom Hersteller vorgegeben wurde, in einen Druckwert umgewandelt. Im nächsten Schritt wird über eine synchrone I2C Anfrage der Bremsenwert des BTDM ermittelt. Alle Messwerte werden an dem lokalen Display angezeigt. Der Raddruck wird an den ROS Topic „/tire_pressure_orig“ gesendet. Der Zusatz „orig“ bedeutet hier dasselbe wie bei den Ausführungen zur ADIRU. Anschließend wird ein Update der ROS internen Zustände durchgeführt, 10 ms gewartet und der Neustart der Schleife initiiert.

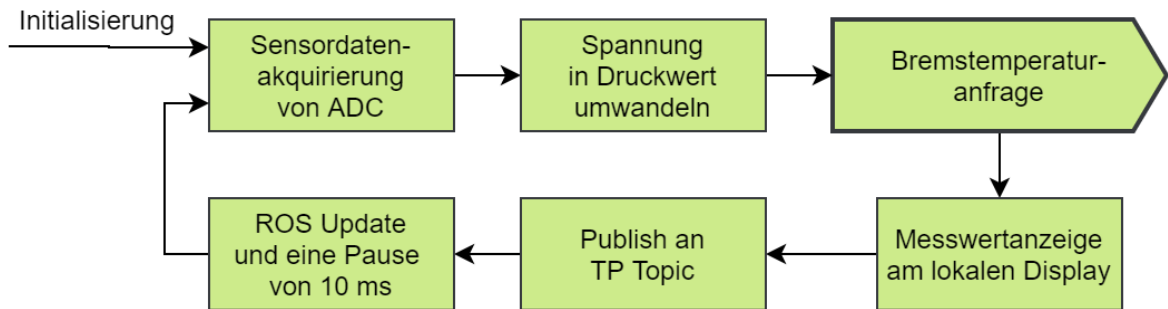


Abbildung 31: Aktivitätsdiagramm des TPDM

In Abbildung 32 ist das Aktivitätsdiagramm des BTDM zu sehen. Nach dem Mikrokontrollerstart werden interne Zustände und die I2C Verbindung zum TPDM initialisiert. Zusätzlich wird der ROS Topic für Bremsenwert deklariert. Die Sensordaten der Bremsenwert werden von dem MAX31865 akquiriert und an den „/braking_temperature_orig“ Topic gesendet. Der Zusatz „orig“ bedeutet hier dasselbe wie die Erklärungen zur ADIRU und der TPDM. Der ROS wird upgedatet und nach 10 ms findet der Neustart der Schleife statt. Wenn eine I2C Anfrage von dem TPDM kommt, wird der Programmfluss unterbrochen und der aktuelle, global gespeicherte Bremsenwert zurück an den TPDM über I2C gesendet.

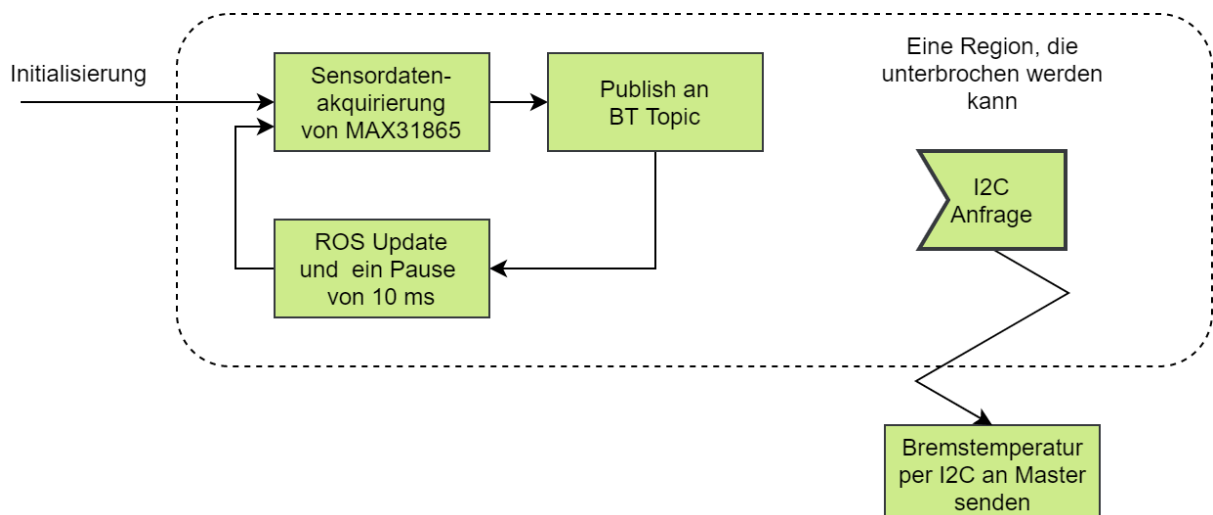


Abbildung 32: Aktivitätsdiagramm des BTDM

5.2.2 Grundlegender Aufbau des verteilten Systems

Elektronische Komponenten und Verdrahtung

In Abbildung 33 ist das Blockschaltbild des verteilten Systems für die Beispielanwendung dargestellt. Fünf Raspberry PI 3B+ dienen als Avionics Computer. Ein zusätzlicher PI wird benötigt, um das System Display ansteuern zu können. Alle Avionics Computer werden mittels Hutschienenadapter an die Hutschiene aufgesetzt.

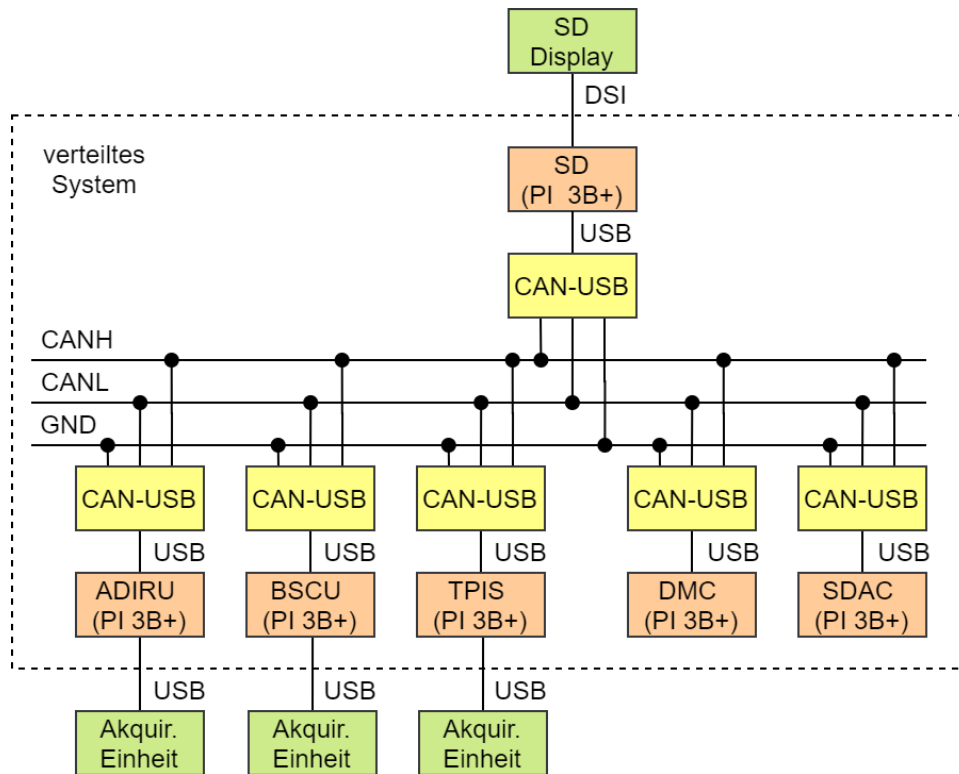


Abbildung 33: Blockschaltbild des verteilten Systems

ADIRU, BSCU und TPIS werden jeweils mit ihren Akquirierungseinheiten verbunden. Für die Kommunikation über das CAN Netzwerk werden alle PI's mit einem CAN-USB Adapter ausgestattet. Die CAN-USB Adapter benötigen auch einen Hutschienenadapter. Diese werden in einem CAD Programm konstruiert und 3D-gedruckt. Alle eingesetzten CAN-USB Adapter werden parallel mit drei CAN Leitungen verbunden. Das sind die CAN High, CAN Low und Masse Leitungen. Die elektrische Verbindung der CAN Leitungen erfolgt über eine Lüsterklemme.

Zusätzlich benötigen alle Raspberry PI's 5 VDC Spannungsversorgung. Diese wird durch den DC-DC Wandler aus 4.4.2 bereitgestellt. Der DC-DC Wandler benötigt einen Hutschienenadapter. Dieser wird konstruiert und 3D-gedruckt. Jeweils ein Raspberry PI wird mit einem DC-DC Wandler ausgestattet, der genau auf 5 VDC eingestellt wird. Der Wandler selbst wird durch die Reihenklemme und die bereitgestellte 24 VDC Spannung des Anwendungsmoduls versorgt. Die Verbindung des Wandlers mit dem PI erfolgt durch ein spezielles Micro-USB Kabel, das mit Schraubklemmen verbunden werden kann. Dadurch kann der PI bei Bedarf verpolungssicher und flexibel angeschlossen und ausgetauscht werden.

Die Abbildung 34 - 36 zeigen das Ergebnis des Einbaus. Jeweils für eine Avionics Gruppe wird die Prozessebene eines Anwendungsmoduls belegt.

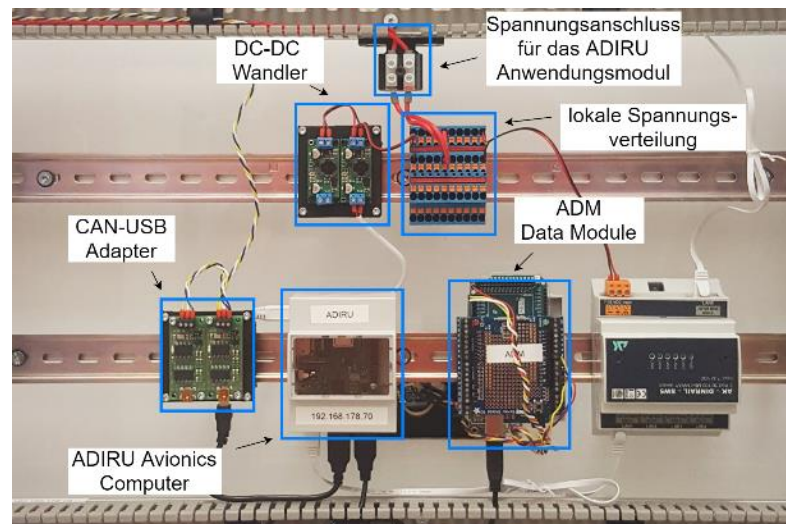


Abbildung 34: Belegung der Prozessebene durch die ADIRU Gruppe

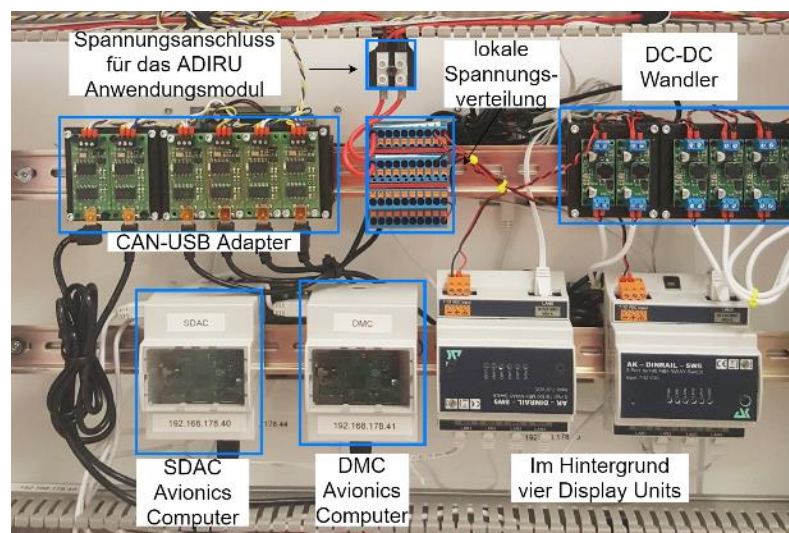


Abbildung 35: Belegung der Prozessebene durch die Cockpit Gruppe (DMC, SDAC, DUs)

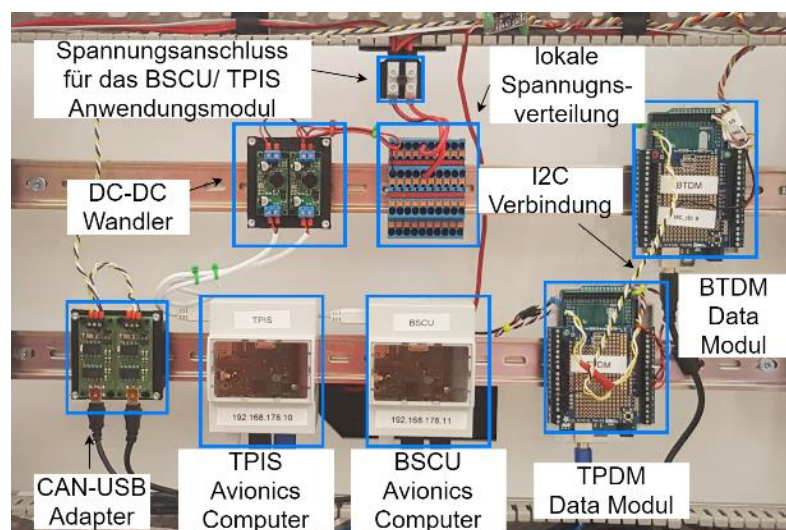


Abbildung 36: Belegung der Prozessebene durch das BSCU/TPIS Gruppe

Betriebssystem und Middleware

Für die Avionics Computer wird die grundlegende Software aufgesetzt. Das ist das Ubuntu Mate Betriebssystem und das ROS Paket, das als Middleware in dieser Arbeit eingesetzt wird. Für die Installation des Betriebssystems wird lediglich die Distribution und ein SD-Kartenadapter benötigt. Es wird Ubuntu Mate 18.04 Long Time Support heruntergeladen. Die Distribution wird auf die SD Karte mittels disk dump (dd) oder Etcher kopiert. Nach dem Kopieren wird die Karte in den PI eingesetzt. Die Installation erfolgt automatisch gleich nach dem Start des Computers. Für die einfache Handhabung wird derselbe Name und dasselbe Passwort für alle Computer gewählt. Die ROS Distribution ist für Ubuntu Mate bereits vorkompiliert, wodurch die Installation einfach durchgeführt werden kann. Es muss lediglich die Quelle des Herstellers in dem Paketmanager von Ubuntu hinzugefügt und die entsprechenden Pakete heruntergeladen werden. Es wird die ROS-Version Melodic installiert, da sie stabil auf Ubuntu Mate 18 ausgeführt werden kann und relativ aktuell ist.

CAN-USB Adapter Autoinitialisierung

Der CAN-USB Adapter dient dem Datenaustausch zwischen den Avionics Computern. Er wird in einer Linux Umgebung im Zusammenspiel mit einem ROS Programm betrieben. In Abbildung 37 ist das Blockschaltbild der funktionalen Kette zu sehen, die benötigt wird, um den Betrieb des CAN Adapters zu ermöglichen. Es werden auf mehreren Ebenen des Betriebssystems spezielle Schnittstellen angesprochen:

- Ein SLCAN Treiber, der den seriellen Austausch zwischen dem CAN-USB Adapter und der SocketCAN Netzwerkabstrahierung ermöglicht. Der Treiber wird durch das Paket „can-utils“ zu Verfügung gestellt.
- Die SocketCAN Abstrahierung, die es ermöglicht, mit dem Socket Layer des Linux Betriebssystems zu kommunizieren. Dadurch kann das CAN Gerät wie ein normales Netzwerkgerät behandelt werden. Zum Beispiel ist es in dem Tool „ifconfig“ sichtbar.
- Das SocketCAN Interface. Die Abstrahierung des Socket Layers ermöglicht den einfacheren Umgang mit der SocketCAN Schnittstelle. Das SocketCAN Interface ist ein Paket von ROS.

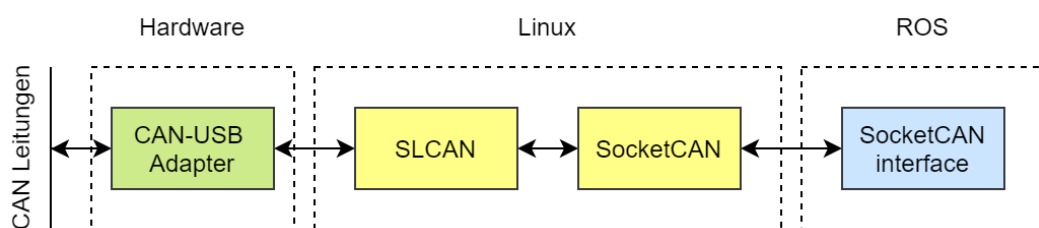


Abbildung 37: Blockschaltbild der Treiberkette für den CAN-BUS Adapter

Die Konfiguration des SLCAN Treibers und die Einbindung in die Netzwerkebene wird standardmäßig nicht durchgeführt. Dieser Prozess soll automatisiert werden durch spezielle „udev“ Regeln und dem systemd Taskmanager. Beim Anschließen und Abstecken der USB Verbindung werden speziell erstellte Skripte ausgeführt, die den SLCAN Treiber initialisieren oder vom System trennen. So kann die Initialisierung des Adapters beim Start des Demonstrators automatisch durchgeführt werden ohne einen weiteren Eingriff des Benutzers zu erfordern.

5.2.3 Kommunikationsmanager für das verteilte System

Der Kommunikationsmanager soll die zentrale Kommunikationsabstrahierung darstellen. Über den Kommunikationsmanager sollen alle Avionics Computer miteinander Daten austauschen.

Damit das System später besser diagnostiziert werden kann, werden zwei Schichten für die Kommunikation definiert:

- CAN Interface – Definition von ein- und ausgehenden Topics für die rohen CAN Pakete
- Channel Interface – Wandlung der CAN Kanäle in einzelne Ein- und Ausgangskanäle von ARINC 429

CAN Interface

Das CAN Interface soll für den einfachen Umgang mit CAN Paketen innerhalb einer ROS Topic Umgebung eingesetzt werden. Durch das CAN Interface sollen zwei Topics zu Verfügung stehen. Das „/can_in“ Topic für das Empfangen von CAN Paketen und „/can_out“ für das Versenden von CAN Paketen.

Für die Implementierung des CAN Interface wird die Programmierschnittstelle von dem SocketCAN Interface verwendet. Beim Ankommen einer neuen CAN Nachricht aus dem physikalischen CAN Netzwerk sendet ein ROS Publisher an den „/can_in“ Topic eine ROS Nachricht mit dem Inhalt des CAN Paketes. Durch das Senden an den „/can_out“ Topic aus der ROS Umgebung wird ein ROS Subscriber angesprochen. Dieser sendet dann mittels der SocketCAN Interface API die CAN Nachricht an das physikalische CAN Netzwerk.

Grundlegende Definition des Channel Interfaces

Das Channel Interface wird benötigt, um die Kommunikation programm-technisch zu abstrahieren und die Verbindungs- und Adressierungsfunktion von ARINC 429 nachzubilden.

In Abbildung 38 ist das generelle Konzept, das hinter einem realen ARINC 429 Adapter steht, zu sehen. Bei ARINC werden für den Nachrichtenaustausch parallele Leitungen von einem Teilnehmer zum anderen verwendet. Jeder Anschluss, sowohl ein Eingang oder ein Ausgang, hat die entsprechende Repräsentierung in der Software als separaten Kanal. Jeder dieser Kanäle kann unterschiedliche Nachrichtentypen versenden, die dann über Labels beim Empfänger unterschieden werden.

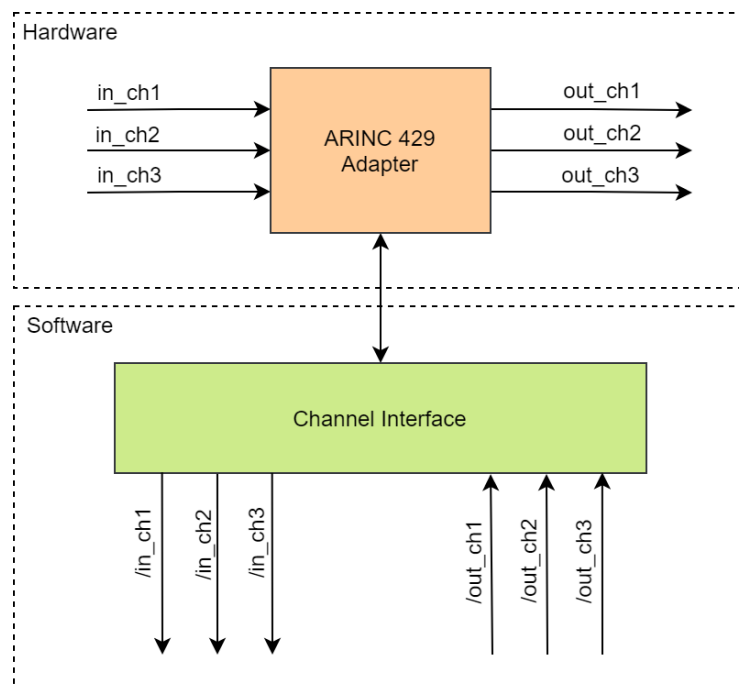


Abbildung 38: Reale ARINC 429 Softwareintegration

Das CAN Protokoll verwendet für jeglichen Nachrichtenverkehr einen einzigen Bus. Damit jedoch die ARINC Adressierungs- und Verbindungsfunktion simuliert werden kann, müssen eine Reihe von Annahmen getroffen werden:

- Jeder Avionics Computer trägt eine eindeutige Machine ID.
- Ein CAN Kanal soll als eine Leitung zwischen zwei ARINC Kommunikationsteilnehmern verwendet werden.
- Für jeden Kommunikationsport wird der Richtungstyp definiert (Eingang oder Ausgang).

In Abbildung 39 ist die generelle Idee dieser Annahmen zu sehen. Für einen Kommunikationsteilnehmer wird eine eindeutige Machine ID und mehrere Ports mit einem gewählten Porttyp definiert. Ein CAN Kanal entspricht dann einer Verbindung über eine Leitung zwischen zwei Kommunikationsteilnehmern.

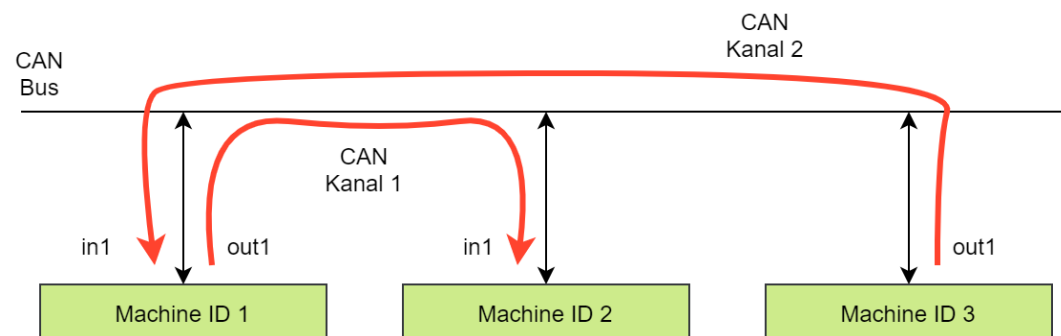


Abbildung 39: Verwendung von CAN für die Implementierung der Adressierungs- und Verbindungsfunktion

Definition von einem ROS Nachrichtentyp für das Channel Interface

Für den Nachrichtenaustausch zwischen zwei Kommunikationsteilnehmern wird ein spezieller ROS Nachrichtentyp definiert. Er soll „ChannelFrame“ bezeichnet werden. Dieser Nachrichtentyp soll ein Label und einen Gleitkommawert aufnehmen können. Die Labels werden in dem Nachrichtentyp für die jeweiligen Werte, die gesendet werden sollen, definiert. In dieser Arbeit werden insgesamt vier Labels definiert:

- Gesamtlufttemperatur
- Statischer Druck
- Bremsstemperatur
- Raddruck

Mit den Labels können die Nachrichten an einen Bus gesendet und nach dem Label gefiltert werden.

Verbindungsmanager für das Channel Interface

Mit den gegebenen Bedingungen wird ein Verbindungsmanager programmiert. Dieser soll mit einer einfach gehaltenen, globalen Verbindungstabelle die Konfiguration selbstständig durchführen können. Die Verbindungstabelle soll an alle Computer durch einfaches Kopieren verteilt werden (Abbildung 40). Die Tabelle soll vor dem Start der entsprechenden Programme bekannt sein.

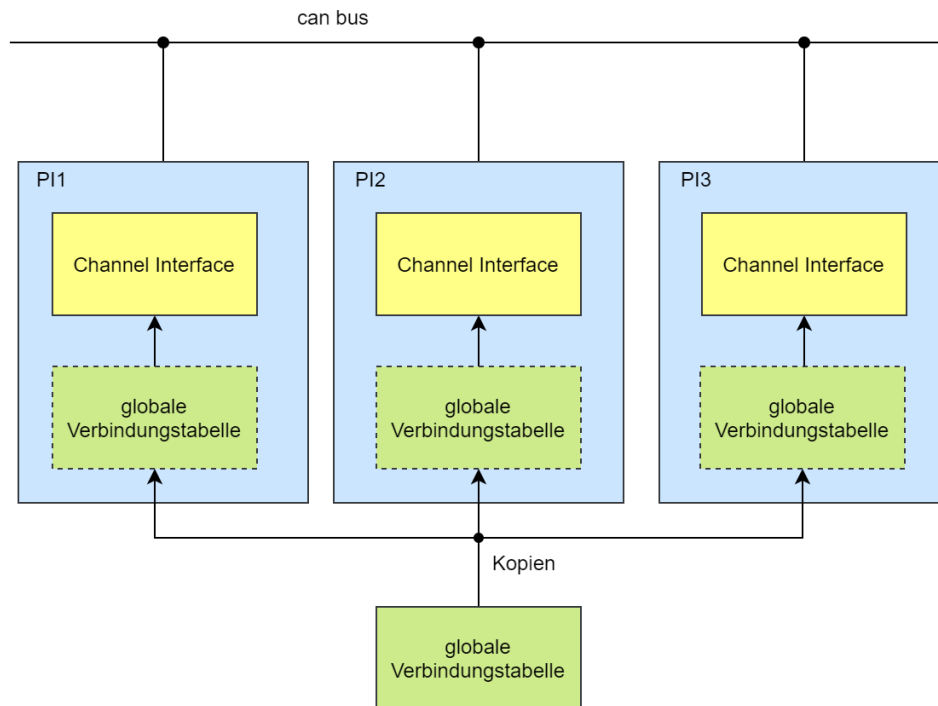


Abbildung 40: Bereitstellung der Verbindungstabelle für Avionics Computer

In Abbildung 41 ist das Aktivitätsdiagramm für die Initialisierung des Verbindungsmanagers zu sehen. Eine globale Verbindungstabelle, die für jeden Avionics Computer gleich ist, wird als Eingabe für das Channel Interface verwendet. Auf der Grundlage dieser Tabelle werden zwei Wörterbücher erstellt, welche Reihennummern, Typen und entsprechende CAN IDs für Ein- und Ausgänge aufnehmen. Mit den Informationen aus den Wörterbüchern und der Machine ID werden Subscriber und Publisher für die jeweiligen Channel Interfaces erstellt. Die Publisher und Subscriber der Channel Interfaces interagieren dann mit den Publishern und Subscribern des CAN Interfaces.

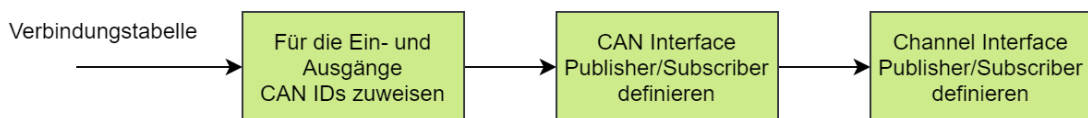


Abbildung 41: Aktivitätsdiagramm für die Initialisierung des Verbindungsmanagers

Beim Senden durch einen Kanal wird die Nachricht serialisiert. Dabei werden die Labelnummer und eine Gleitkommazahl von der „ChannelFrame“-Nachrichtstruktur in ein fünf Byte Array umgewandelt. Das erste Byte ist das Label, die nachfolgenden vier Bytes entsprechen der Gleitkommazahl in Byteform. Beim Empfangen werden diese zwei Werte wieder getrennt und in eine „ChannelFrame“-Struktur kopiert.

Die Zusammenfassung zu dem Kommunikationsmanager

Das CAN Interface und das Channel Interface werden zu einem Kommunikationsmanager zusammengefasst. Der Kommunikationsmanager beinhaltet die globale Verbindungstabelle und muss für alle Kommunikationsteilnehmer gleich sein.

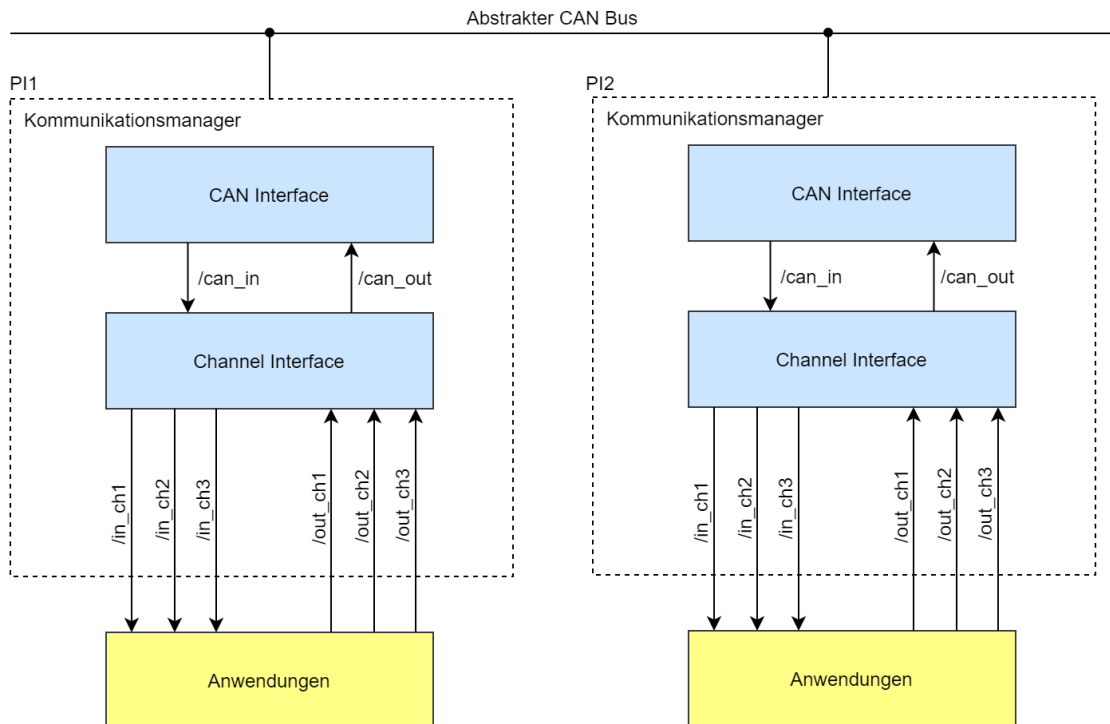


Abbildung 42: Blockschaltbild für die beispielhafte Anwendung des Kommunikationsmanagers

In Abbildung 42 ist ein Beispiel für die Anwendung des Kommunikationsmanagers zwischen zwei Raspberry Pis zu sehen. Die Abbildung 43 zeigt eine mögliche Verbindungstabelle an. Es werden durch drei Einträge eine Reihe von Verbindungen zwischen dem PI1 und PI2 hergestellt.

```
connections:
- { "PI1" : "out_ch1", "PI2" : "in_ch1" }
- { "PI1" : "out_ch1", "PI2" : "in_ch2" }
- { "PI2" : "out_ch1", "PI1" : "in_ch1" }
```

Abbildung 43: Beispielverbindungstabelle

Innerhalb der geschweiften Klammern eines Eintrages wird eine Verbindung definiert: Links vor dem Komma wird immer der Sender definiert, rechts nach dem Komma der Empfänger. Der Name links vor dem Doppelpunkt definiert die Machine ID. Rechts vor dem Doppelpunkt werden der Richtungstyp und die Reihennummer definiert. Es können auch mehrere Eingänge mit einem Ausgang verbunden werden.

5.2.4 Die Zusammenführung

Alle bisher erarbeiteten Grundbausteine werden dafür genutzt, um die geforderte Struktur aus Abbildung 19 implementieren zu können.

Die Struktur

In Abbildung 44 ist die aktuelle Verbindungsstruktur des Demonstrators zu sehen. Die TPIS Akquirierungseinheit nimmt den Raddruck-Wert auf und sendet ihn über den „tire_pressure_orig“ Topic an die TPIS. Die TPIS baut mit dem Messwert ein ChannelFrame mit

dem „Tire Pressure“ Label. Über den „out1“ Kanal wird dieser Wert an den SDAC gesendet. Die BSCU Akquirierungseinheit nimmt den Bremsdruckwert auf und sendet ihn über den „/braking_temperature_orig“ Topic an die BSCU. Die BSCU baut mit dem Messwert ein ChannelFrame mit dem „Braking Temperature“ Label. Über den „out1“ Kanal wird dieser Wert an den SDAC gesendet. Die ADIRU Akquirierungseinheit nimmt die Werte für den statischen Luftdruck und die Gesamtlufttemperatur auf und sendet sie über die Topics „/static_air_pressure_orig“ und „/total_air_temperature_orig“ an die ADIRU. Die ADIRU baut mit dem Messwert zwei ChannelFrames mit den „Static Pressure“ und „Total Air Temperature“ Labels. Über den „out1“ Kanal werden diese Werte an den DMC gesendet. Die SDAC sendet alle ihre empfangenen Werte an den DMC. Die DMC sendet alle ihre empfangenen Werte an das System Display. Das System Display zeigt über das rqt System und der darunter liegenden PyQt Implementierung alle empfangenen Werte auf einer grafischen Oberfläche an.

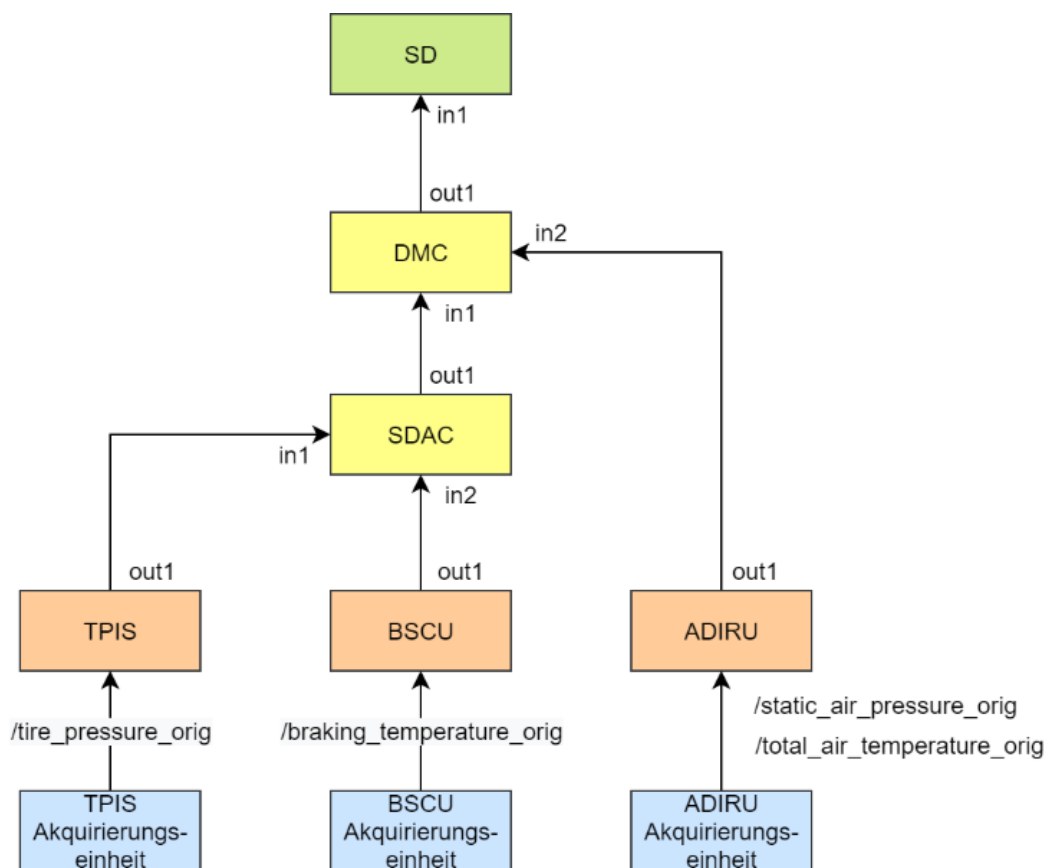


Abbildung 44: Die Implementierung der geforderten Verbindungsstruktur

Die Avionics Computer werden mittels des Verbindungsmanagers und einer globalen Verbindungstabelle „verdrahtet“. Die Abbildung 45 zeigt die aktuell verwendete Verbindungstabelle der Avionics Computer.

```

connections:
#tpis output
- { "tpis": "out_ch1", "sdac": "in_ch1" }

#bscu output
- { "bscu": "out_ch1", "sdac": "in_ch2" }

#sdac output
- { "sdac": "out_ch1", "dmc": "in_ch1" }

#adiru output
- { "adiru": "out_ch1", "dmc": "in_ch2" }

#dmc output
- { "dmc": "out_ch1", "sd": "in_ch1" }

```

Abbildung 45: Verbindungstabelle des aktuellen Aufbaus

Der Autostart

Damit alle Computer nach dem Start die jeweiligen Programme automatisch hochfahren, werden zwei Ansätze verwendet. Das sind das ROS Hilfsprogramm „Robot-Upstart“ und das Linux Hilfsprogramm „Startup Applications“.

Robot-Upstart verwendet im Hintergrund den systemd Taskmanager. Wegen der Initialisierungsreihenfolge von Linux können nur Programme ohne eine grafische Oberfläche gestartet werden. Die ROS Programme von BSCU, TPIS, ADIRU, SDAC, DMC werden mit Robot-Upstart für den Autostart konfiguriert.

Startup Applications verwendet im Hintergrund das X System und ist für Programme mit einer grafischen Umgebung entwickelt worden. Ein spezielles Skript muss erstellt werden, um das Starten des ROS Programms für das System Display zusammen mit der grafischen rqt Oberfläche zu ermöglichen. Das ist dadurch bedingt, dass noch zusätzliche Schritte vor dem eigentlichen ROS Start durchgeführt werden müssen, um ROS erfolgreich ausführen zu können. Dieses Skript wird durch das Hilfsprogramm Startup Applications von Ubuntu für den automatischen Start konfiguriert.

6 Test

Dieses Kapitel beschäftigt sich mit der Untersuchung und der Bewertung der realisierten Konzepte und Gerätschaften.

Der Test wird in drei Etappen durchgeführt. Zuerst wird nur die Beispielanwendung untersucht (Abschn. 6.1), dann die Demonstratortechnik selbst (Abschn. 6.2). Bei der Beispielanwendung und der Demonstratortechnik wird bewertet, inwiefern die gestellten Spezifikationen erfüllt werden. Schließlich wird ein allgemeiner Anwendungsfall formuliert, der auf den vorherigen zwei Punkten basiert (Abschn. 6.3). Der allgemeine Anwendungsfall hilft zu bewerten, ob das Gesamtziel dieser Arbeit erfüllt worden ist.

Als Grundlage für den Test dienen die Anforderungstabellen Tabelle 1, Tabelle 2 und Tabelle 3. Nicht über alle Anforderungen lässt sich eine Diskussion führen. Bei einigen Anforderungen kann nur entschieden werden, ob eine Eigenschaft erfüllt wird oder Komponenten existieren. Entsprechend werden einige dieser Anforderungen nicht explizit erwähnt und sind nur in den Anforderungstabellen wiederzufinden. Abhängig davon, ob die betrachtete Anforderung erfüllt wurde, sind alle Anforderungen in den genannten Tabellen in der Spalte „Erfüllt“ mit einem „Ja“ oder „Nein“ gekennzeichnet.

6.1 Beispielanwendung

Die Beispielanwendung wird auf Grundlage der in 3.2 formulierten Spezifikation bewertet. Untersucht wird dabei der umgesetzte Messaufbau und das Kommunikationsnetz der Avionics Computer.

6.1.1 Beispielmessung

Zuerst wird verglichen, ob die Messwerte auf den lokalen und entfernten Display Units übereinstimmen. Beide abgebildeten Werte werden zeitgleich aktualisiert. Durch eine einfache Betrachtung entsteht so gut wie keine registrierbare Verzögerung.

Es werden die erwarteten Ruhe- und Mittelwerte sowie Messbereiche der Sensoren untersucht. Die Sensoren werden primär in einer Labor- bzw. Ausstellungsumgebung betrieben. Somit entsprechen ihre Normalwerte den normalen Raumbedingungen.

Tabelle 26: Vergleich des erwarteten Normalwerts und einer Momentaufnahme der Messwerte

	Statischer Luftdrucksensor in mbar	Gesamtluft- temperatursensor in Grad	Raddrucksensor in mbar	Bremstemperatur- sensor in Grad
Erwarteter Normalwert	1013,25	20	0	20
Moment- aufnahme	1014,93	23,39	27,34	19,20

Die Tabelle 26 zeigt den Vergleich zwischen dem erwarteten Normalwert und einer Momentaufnahme der Messwerte. Die 1013,25 mbar sind ein allgemein bekannter Wert für Normaldruck der Atmosphäre. Bei dem Gesamtlufttemperatursensor wird die aktuelle

Umgebungstemperatur gemessen. Diese liegt im Schnitt bei 20 Grad bei normalen Raumbedingungen. Um zu verifizieren, dass die aktuellen Messwerte des ADIRU Systems immer die erwarteten Werte darstellen, wird der Demonstrator über mehrere Tage angeschaltet gelassen und dabei die Messwerte aufgezeichnet. Die in Tabelle 26 aufgezeichneten Momentaufnahmen repräsentieren den Mittelwert der durchgeführten Beobachtung. Die Messwerte für statischen Luftdruck und Gesamtlufttemperatur unterscheiden sich leicht durch lokale Wetterschwankungen.

Der Messwert für den Raddrucksensor wird mit einem Prozessluftschlauch und einer einfachen Pumpe einer Medizinspritze getestet. Der Raddrucksensor misst relativ zum Normaldruck. Bei dieser Messung entsprechen 0 mbar dem Normaldruck von 1013,25 mbar. Die Momentaufnahme in Tabelle 26 zeigt einen Wert, der sich leicht von dem erwarteten Druck unterscheidet. Das ist dadurch bedingt, dass die Pumpe beim Loslassen nicht exakt wieder die Nulllage einnimmt. Als Schwierigkeit hat sich gezeigt den Sensor über seine Messbereichsgrenze von 4 bar hinaus zu bringen, was für Vorführungen vorteilhaft ist. Dadurch werden die Betrachter eventuell länger mit diesem Vorgang beschäftigt sein.

Der Messwert für den Bremstemperatursensor wird mit einem Bunsenbrenner getestet. In Abbildung 46 ist ein Temperaturverlauf durch das Erhitzen mit dem Bunsenbrenner zu sehen. Wie erwartet hat der Sensor eine hohe Trägheit. Er kann sehr schnell erhitzt werden und nur sehr langsam abkühlen. Da der Bunsenbrenner eine sehr hohe Dynamik erzeugt, kann der vorgesehene Messbereich sehr schnell verlassen werden. Es sollte darauf geachtet werden, dass der Messwert 200 Grad nicht übersteigt.

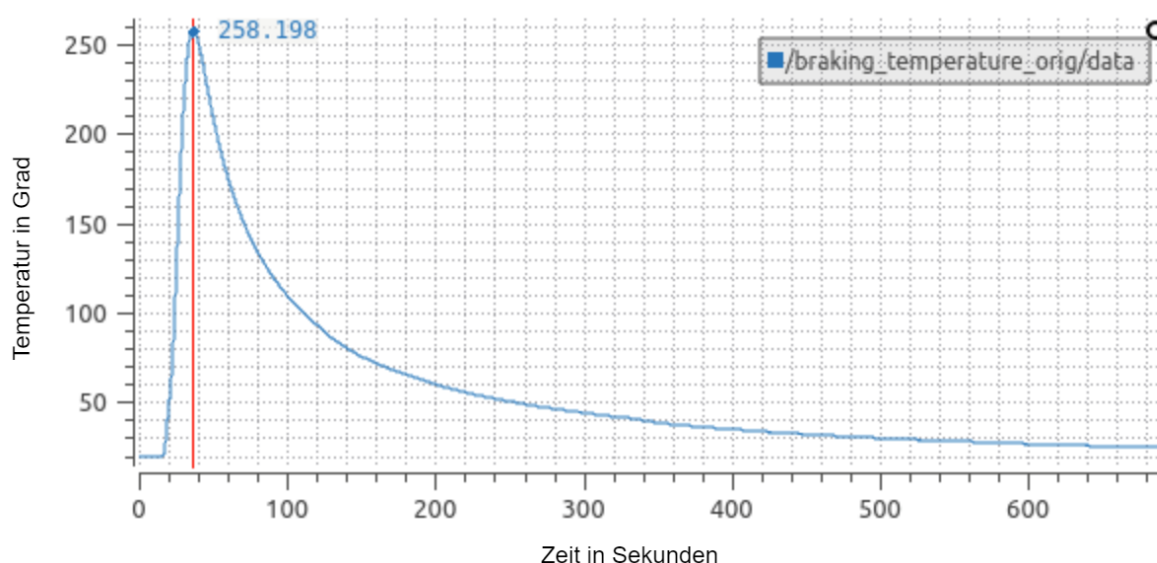


Abbildung 46: Bremstemperaturtest

6.1.2 Kommunikation

Abstrahierung der Kommunikation

Kommunikationsverbindung sollte entsprechend den Anforderungen abstrahiert werden. Es sollte weiterhin möglich, sein Computer auf eine ARINC ähnliche Weise zu verbinden. Die Pakete sollten über Labels adressiert werden können. Diese Anforderungen wurden im Grundsatz erfüllt: Die aktuelle Implementierung ermöglicht es über CAN-BUS und einen Verbindungsmanager in der Software, jegliche Computer untereinander zu verbinden. Die Kopplung an die

Software geschieht über standardisierte Kommunikationsschnittstellen. Diese Verbindungsart hat den Vorteil einer übersichtlichen Kabelführung und schnellen Verdrahtung unter den Computern und einer einfachen Verbindung der Softwareanwendung mit dem Kommunikationsnetz. Nachteilig ist die fehlende Möglichkeit reale ARINC Geräte anschließen zu können. Die Nachbildung ist jedoch ein essentieller Schritt, um auf die reale ARINC Hardware umzusteigen. Für volle Kompatibilität mit ARINC muss dabei lediglich das „Channel Interface“ in der Software neu implementiert, der Rest der Software aber nicht verändert werden.

Latenz in der Kommunikation

Das Aktualisierungsintervall der Messwerte an den Display Units sollte nicht kürzer als 62,5 ms sein. Somit darf die Latenzzeit der Kommunikation nicht höher als das Aktualisierungsintervall ausfallen. So ist in erster Linie von Interesse, wieviel Zeit das Signal von einer Akquirierungseinheit zur API der grafischen rqt-Oberfläche auf dem System Display benötigt. Um dies zu untersuchen, wird die längste Systemkette betrachtet, da alle anderen Ketten demnach kürzere Latenzzeiten haben werden.

Die Abbildung 47 zeigt das Konzept und den Aufbau der Messung. Die längste Systemkette wird durch das TPIS System gebildet. Die gesamte Systemkette besteht aus dem TPDM, TPIS, SDAC, DMC und dem SD. Für die Latenzmessung werden der Moment, an dem der Messwert akquiriert und der Moment, an dem der Messwert auf dem PI Display angezeigt wird, fixiert. Somit können die TPIS, SDAC, DMC Avionics Computer zu einer Zwischenkette zusammengefasst werden.

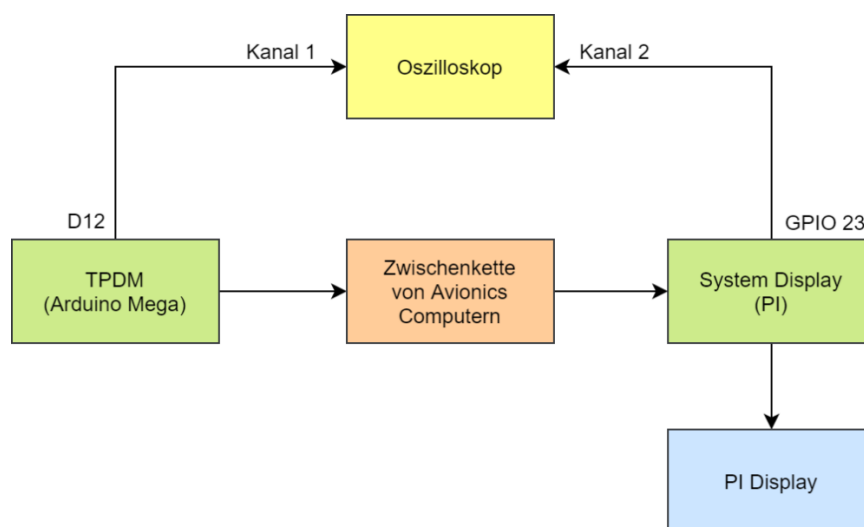


Abbildung 47: Latenzmessung mit Oszilloskop

Die Messung wird mit einem Oszilloskop durchgeführt. Der TPDM Arduino Mikrokontroller schaltet den D12 Ausgang auf 5 V, wenn ein neuer Messwert vorliegt. Kurze Zeit später wird der D12 Ausgang wieder auf 0 V geschaltet. Gleich nach dem Ankommen in der GUI auf dem System Display, wird der GPIO 23 von dem PI auf 3,3 V geschaltet. Kurze Zeit später wird er wieder auf 0 V resettet. Die Zeit zwischen den vorderen Flanken der Pulse des Arduino und des PI bilden die Latenzzeit. Durch das Triggern im Oszilloskop auf Kanal 1 können Signale der beiden Geräte (Arduino und PI) synchronisiert werden. Die Zeitdifferenz wird durch die Cursor Funktion des Oszilloskops ermittelt.

Die Messung ergibt zwei Grenzwerte:

- Minimalverzögerung: 32 ms
- Maximalverzögerung: 38 ms

Die Latenzzeit liegt also weit unter den geforderten 62,5 ms des Aktualisierungsintervalls der Messwerte auf den Display Units.

6.2 Demonstratortechnik

In diesem Unterkapitel wird bewertet, inwiefern die Demonstratortechnik die geforderte Spezifikation erfüllt. Hierbei dient die Integration der Beispielanwendung als Grundlage für die Bewertung.

6.2.1 Präsentierbarkeit

Die Abbildung 48 zeigt den umgesetzten Aufbau des Demonstrators. Zu sehen sind die visuellen Einheiten der ADIRU und BSCU/TPIS Akquirierungseinheiten, vier Display Units und zusätzlich unterstützende Grafiken. Die Grafiken sollen die Ergebnisse der Messung zeigen.

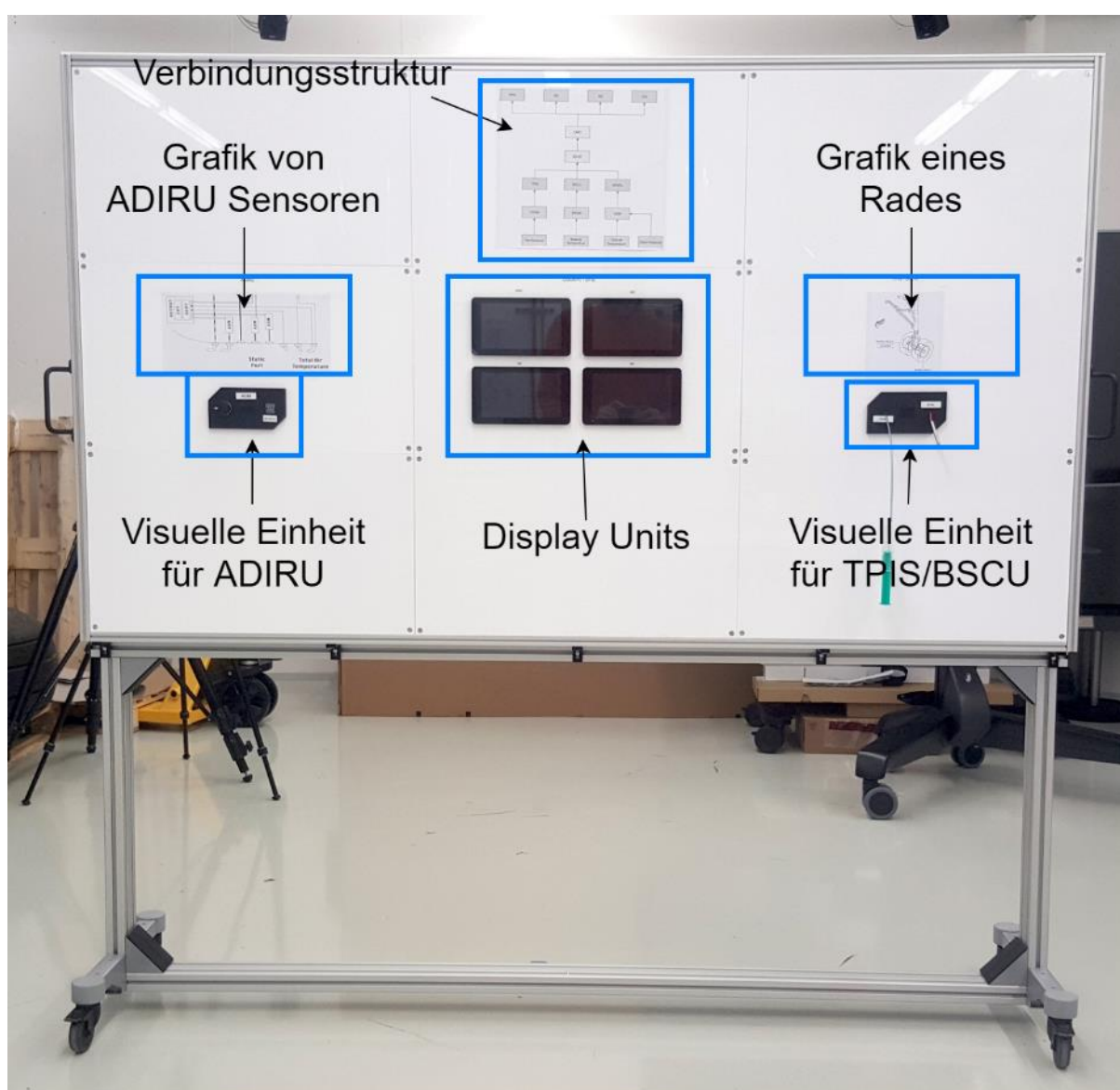


Abbildung 48: Umgesetzter Aufbau des Demonstrators

Die Abbildung 49 zeigt die grafische Oberfläche auf dem System Display. Die Grafiken, die auf der Präsentationsfläche aufgeklebt sind, werden hier noch einmal optisch dargestellt und stellen eine inhaltliche Verbindung für den Betrachter dar.

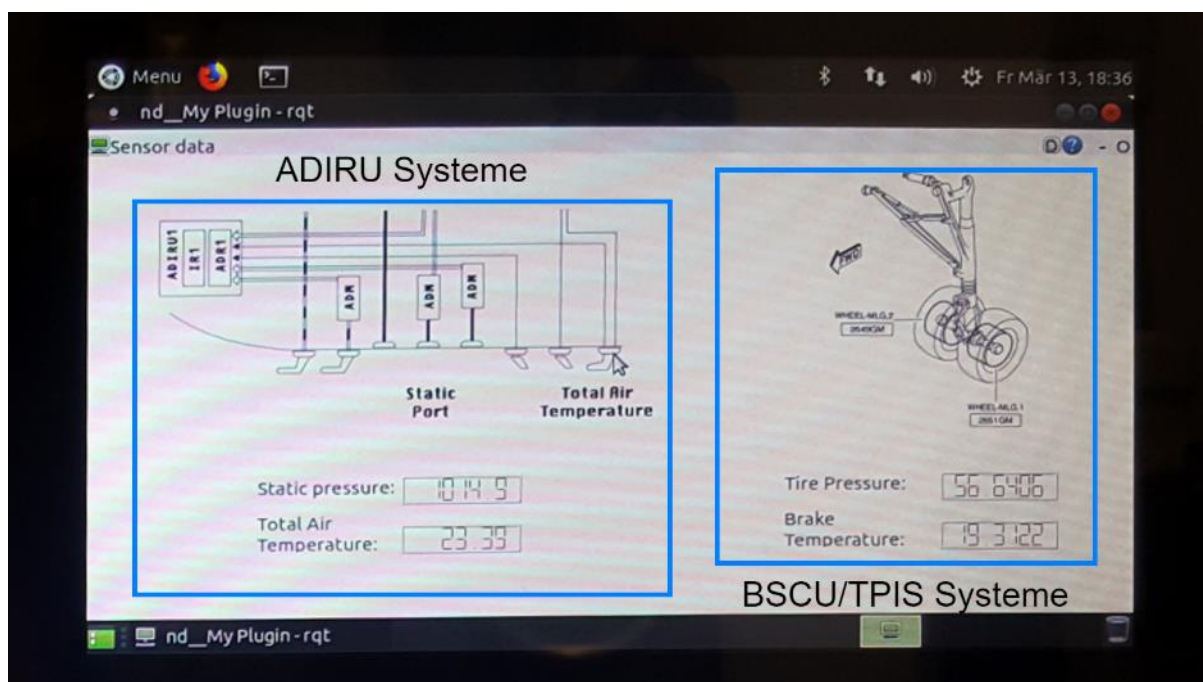


Abbildung 49: Grafische Oberfläche des System Display

Um die Wirkung der Optik auf das ästhetische Gefühl der Kollegen vom DLR zu testen, wird ihnen der Demonstratoraufbau vorgeführt. Es werden hierbei zwei Fragen gestellt:

- Wie wirkt das äußerliche Design?
- Ist es klar, was gemessen und dargestellt wird?

Bei allen Befragten blieb ein positiver Eindruck vom Design und Aufbau. Die visuellen Elemente der Beispielanwendung wirkten auf sie zusammengehörend und klar strukturiert. Es wurde deutlich, dass die Messungen die Zustände des Flugzeuges darstellen. Durch das Spiel mit dem Bunsenbrenner und der Pumpe konnten die Kollegen einen besseren Eindruck dafür bekommen, was mit dem Demonstrator gemacht werden kann.

6.2.2 Erweiterbarkeit

Es wird untersucht, welche Schritte durchgeführt werden müssen, um neue Technologien in den Demonstrator zu integrieren. Dabei wird die Beispielanwendung als Grundlage der Untersuchung dienen. Bei der Erweiterung wird angenommen, dass ein neues Flugsystem integriert werden kann. Dieses Flugsystem soll Daten messen und mit dem Rest der Computer kommunizieren können. Als Beispiel wird das ADIRU System betrachtet.

Für das ADIRU System wird ein Anwendungsmodul des Demonstrators gewählt. Der Avionics Computer und die zusätzliche Elektronik werden mit einer Hutschiene innerhalb des Anwendungsmoduls befestigt. Für das Anwendungsmodul wird auf der Präsentationsebene eine Acrylplatte mit vorgefertigten Bohrungen genommen. Diese Platte wird mit ADIRU Sensoren belegt. Über eine Reihenklemme können innerhalb des Anwendungsmoduls 24 VDC an die Elektronik verteilt werden. Ein DC-DC Wandler stellt dem Avionics Computer und der Zusatzelektronik die erforderliche Versorgungsspannung zur Verfügung. Alle weiteren elektrischen

Verbindungen werden über Schraubklemmen bzw. verpolungssichere Stecker durchgeführt. Für die Kommunikationsverbindung wird ein CAN-USB Adapter an den Avionics Computer und an den CAN-Bus elektrisch angeschlossen. Der Verbindungsmanager in der Software der Avionics Computer wird angepasst. Dadurch kann die erforderliche Kommunikationsverbindung zwischen den Flugsystemen hergestellt werden. Das Anwendungsprogramm stellt eine Verbindung über Publisher und Subscriber zu einer abstrakten Schnittstelle her, die durch den Verbindungsmanager initialisiert wird.

Reihenklamme, Hutschiene und der Verbindungsmanager ermöglichen einen schnellen und flexiblen Einbau neuer elektronischer Komponenten. Anstatt Bohrungen für Befestigungen selbst vorzunehmen, wird die Hutschiene verwendet. Anstatt zu löten wird geklemmt. Anstatt eine Softwareänderung in ihrem Kern vorzunehmen, wird lediglich nur ein Teil dazu gefügt.

6.2.3 Autarkie

Die elektronischen Geräte sollen ohne externe Stromversorgung in den Demonstrator integriert werden. Hierbei soll untersucht werden, wie viele weitere Systeme in den Demonstrator ohne den Umbau der aktuellen Stromversorgung integriert werden können. Nach der Betätigung des Hauptschalters sollen sich alle elektronischen und programm-technischen Komponenten für einen definierten Anwendungsfall selbstständig innerhalb von zwei Minuten initialisieren. Hierbei soll untersucht werden, ob der Demonstrator immer deterministisch in einen definierten Zustand übergeht.

Stromverbrauch

Das Hauptnetzteil sollte bis zu 15 Avionics Computer mit zusätzlicher Elektronik tragen versorgen können, wobei eine Stromgrenze von 10 A für das Hauptnetzteil in 4.4.2 ermittelt worden ist. Zusätzlich sollte der ermittelte Kabelquerschnitt den Strom bis zur Stromgrenze führen können. Das Kabel für die Energieverteilung hat einen Querschnitt von 1 mm² und kann ebenfalls mit 10 A belastet werden. Es soll also untersucht werden, wie hoch der aktuelle Stromverbrauch ist und wie viele neue Geräte bei der aktuellen Stromgrenze noch versorgt werden können.

Die Beispielanwendung benötigt Strom für neun Avionics Computer, neun CAN-USB Adapter, drei Akquirierungseinheiten und vier Display Units. Die Avionics Computer haben einen unterschiedlichen Stromverbrauch beim Hochfahren, beim Kompilieren und im Normalbetrieb. Die Abbildung 50 zeigt den Messaufbau für die Strommessung. Der Multimeter ist in Reihe geschaltet, er befindet sich in dem Modus für Strommessung. Für alle drei Betriebsfälle wird der Gesamtstrom I_g gemessen. Die Tabelle 27 zeigt das Ergebnis der Messung.

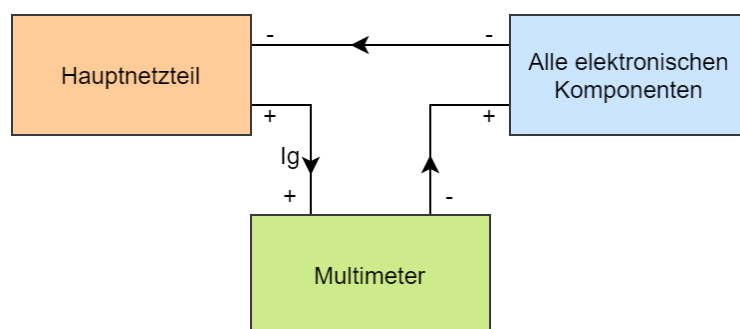


Abbildung 50: Messaufbau für die Gesamtstrommessung

Tabelle 27: Gesamtstrom bei drei Betriebsfällen

Betriebsfall	Gesamtstrom Ig in A
Hochfahren	3,5
Kompilieren	3,0
Normalbetrieb	2,3

Von allen drei Betriebsfällen verbraucht das Hochfahren am meisten Strom; dies ist folglich zu berücksichtigen. Das Netzteil wird hierbei nur zu 1/3 belastet. Mit der Annahme, dass jede andere Flugsystemnachbildung 0,388 A an Strom benötigt, könnten insgesamt bis zu 25 Avionics Computer versorgt werden. Es gibt also bei dem jetzigen Verbrauch noch Luft für 16 weitere Computer.

Selbstständige Initialisierung

Der Demonstrator soll nach der Betätigung des Hauptschalters unter 2 Minuten selbstständig hochfahren und jegliche Komponenten initialisieren können. Für das selbstständige Initialisieren in einer definierten Zeit ist in erster Linie eine deterministische Bootreihenfolge verantwortlich. Jede Softwarekomponente wird in derselben Reihenfolge und in derselben Zeit initialisiert. Wenn der Demonstrator nach zehn Hochfahrversuchen im Schnitt dieselbe Zeit benötigt wird angenommen, dass der Demonstrator immer deterministisch hochfahren wird.

Die Zeitmessung erfolgt mit einer Stoppuhr. Bei der Messung ergibt es Sinn, die Messwerte auf die Sekunde genau zu runden. Als vollständig hochgefahren gilt der Zustand, in dem sich die Sensorwerte auf dem System Display anfangen zu aktualisieren. Im Schnitt ist die Anlage nach 77 Sekunden vollständig hochgefahren. Die Abweichungen von +/- einer Sekunde sind auf die Ungenauigkeit durch das Messen mit der Stoppuhr zurückzuführen.

6.2.4 Verfügbarkeit

Die Komponenten sollten gut dokumentiert sein und Anwendungsbeispiele mitliefern. Weiterhin sollten sie wirtschaftlich sein und eine Langzeitunterstützung besitzen.

Im Demonstrator werden Produkte eingesetzt, die in der Open-Source Hobbywelt weit verbreitet sind. Das sind Raspberry PI's, Arduinos und andere quelloffene Technologien. Die Offenheit für Diskussionen seitens der Community und die Weiterentwicklung durch den Hersteller haben zur Folge, dass Neueinsteiger viel einfacher und schneller eine Technologie in ihrer Anwendung einsetzen können. Dadurch steigern sich auch die Möglichkeiten für eine Integration, da an viel mehr Stellen Veränderungen durchgeführt werden können. Im Gegensatz dazu hätten nicht quelloffene Technologien die Ausbremsung von Entwicklungsprozessen zur Folge. Es wurden Technologien gewählt, die breit verfügbar und nicht luftfahrtspezialisiert sind. Dadurch wurden die Kosten automatisch geringgehalten.

6.2.5 Grenzmaße/Grenzwert

Bei dem Aufbau müssen Grenzmaße und das Grenzwert beachtet werden. Der Demonstrator sollte höchstens 2 m hoch, unter 2 m breit und 0,2 m tief sein. Das Gesamtgewicht dürfte 100 kg nicht übersteigen. Das Gerüst sollte zusammen mit den Außenabdeckungen unter 70 kg wiegen, damit für die Elektronik 30 kg Restgewicht bleibt. Zu prüfen ist nun, welcher Bauraum und welches Gewicht tatsächlich belegt wird und wie viel von diesen Parametern für zukünftige Anwendungen noch zur Verfügung steht.

Gesamtmaße

Bei den Gesamtabmessungen werden zwei Aspekte betrachtet: die Maße des gesamten Aufbaus und die Maße der Präsentationsfläche, die durch die Maße des Obergestells geprägt wird. Die Messung wird mit einem Zollstock durchgeführt. Als erstes werden die Maße der Präsentationsfläche untersucht. Es wird die ursprünglich angenommene Vorgabe, die Anpassung an die Optik und an die aktuellen Maße verglichen. Die Tabelle 28 zeigt das Ergebnis der Messung der Präsentationsfläche. Die aktuellen Maße unterscheiden sich von den ursprünglich angenommenen. Die Abweichung ist durch die design-technische Anpassung bedingt, kann aber vernachlässigt werden, da sie für den Gesamtaufbau nicht entscheidend ist. Es ist außerdem vorteilhaft, dass Bildschirme, die ein 16:9 Seitenverhältnis haben, innerhalb der Präsentationsfläche verbundener wirken.

Tabelle 28: Vergleich der Maße der Präsentationsebene

	Ursprünglich angenommene Maße in mm	Angepasste Maße in mm	Aktuelle Maße in mm
Höhe	1000	1080	1080
Breite	2000	1920	1920

Als nächstes werden die Gesamtmaße betrachtet. Es werden die ursprünglichen Vorgaben, die Anpassung an die Optik / den Türrahmen und die aktuellen Maße verglichen. Die Tabelle 29 zeigt das Ergebnis der Messung des Gesamtaufbaus. Die aktuellen Maße unterscheiden sich von den ursprünglichen Angaben. Das ist durch die design-technische Anpassung innerhalb der Präsentationsfläche und durch die Anpassung der Gesamthöhe an die standardmäßigen 2 m hohen Türrahmen bedingt. Der Unterschied ist aber minimal und kann vernachlässigt werden. Weiterhin bringt die verminderte Gesamthöhe den Vorteil, dass keine Probleme beim aufrechten Transport durch einen Türrahmen entstehen.

Tabelle 29: Vergleich der Gesamtmaße des Demonstrators

	Ursprünglich vorgegebene Maße in mm	Angepasste Maße in mm	Aktuelle Maße in mm
Höhe	2000	1900	1905
Breite	2000	2000	1960
Tiefe	200	200	200

Maße der Anwendungsmodule

Der Demonstrator besitzt neun Anwendungsmodule. Diese können durch eine Hutschiene mit verschiedener Elektronik belegt werden. Es wird angenommen, dass die Prozessebene von einem Anwendungsmodul für die Stromversorgung und die globale Hilfelektronik reserviert wird. Durch das Design steht für ein Anwendungsmodul ein Bauraum von 640 mm x 360 mm x 150 mm zur Verfügung. Es wird angenommen, dass für die meisten Anwendungen zwei Hutschienen in ein Anwendungsmodul eingebaut werden. Weiter wird angenommen, dass eine Hutschiene immer für Avionics Computer belegt wird und eine für Versorgungs- und Hilfelektronik. Bei den gegebenen Maßen können bis zu 5 Raspberry PI's auf eine Hutschiene aufgesetzt werden. Bei 8 Anwendungsmodulen können im Idealfall insgesamt 40 Raspberry PI's eingesetzt werden.

In der Beispielanwendung werden drei Anwendungsmodule auf der Prozess- und Präsentationsebene belegt. Somit stehen aktuell noch 5 Anwendungsmodule für weitere Zwecke zur Verfügung.

Gesamtgewicht

Bei den Maßen gibt es zwei Begrenzungen. Einerseits das Gesamtgewicht, andererseits das Gewicht, das der Elektronik mindestens zur Verfügung stehen muss.

Das Gesamtgewicht kann durch zwei Waagen gemessen werden. Es wird hierbei eine Gleichverteilung des Gewichtes angenommen. Dadurch wird eine Waage genau die Hälfte des Gewichtes zeigen. Durch Addition der beiden Gewichte kann das Gesamtgewicht ermittelt werden. Die Messung ergibt ein Gewicht von 78,31 kg. Somit können ungefähr noch 22 kg mit weiterer Elektronik und Mechanik belegt werden.

Bei der Messung des Gerüsts und der Abdeckungen muss eine Schätzung durchgeführt werden, da der Abbau und Wiederaufbau der gesamten Elektronik zu aufwendig wäre. Die Tabelle 30 zeigt das Ergebnis der Schätzung. Es stehen ungefähr 33 kg für die gesamte Elektronik zur Verfügung. Bei einem Gewicht von einem Raspberry PI mit einem Hutschienengehäuse von ungefähr 300 Gramm würden 110 PI's verbaut werden können.

Tabelle 30: Schätzung des Gewichtes der tragenden und abdeckenden Mechanik

Komponentengruppe	Gewicht
Aluprofile	30 kg
Montagezubehör	7 kg
Abdeckungen	30 kg
Gesamt	67 kg

Die Differenz (siehe oben) zeigt deutlich, dass für die Beispielanwendung aktuell 11 kg benötigt werden. Somit wird das Gewicht für die Elektronik nur zu einem Drittel belegt.

6.2.6 Personen- und Geräteschutz

Der Aufbau soll kippstabil sein, geerdet und mit einer Komponentenspannung von maximal 60 VDC betrieben werden. Es sollen Sicherungen für die Sekundär- und Primärseiten des Wandlers eingebaut werden.

Kippstabilität

Der Demonstrator soll beim versehentlichen Anlehnen an den Aufbau nicht kippen. Es wird angenommen, dass sich eine fiktive Person mit dem Rücken an die Mitte des Aufbaus angelehnt hat. Von allen Punkten des Aufbaus ist diese Stelle potenziell am wenigsten kippsicher. Für den Test werden zwei Kontaktpunkte definiert: Ein Punkt liegt genau in der Mitte des Aufbaus und definiert die Normallage. Ein zweiter Punkt liegt am oberen Ende und definiert die höhere Lage. Die Abbildung 51 zeigt die zwei Kontaktstellen.

Es werden drei Fälle betrachtet:

- Schieben in der Normallage
- Schieben in der höheren Lage
- Anlehnen mit dem Rücken

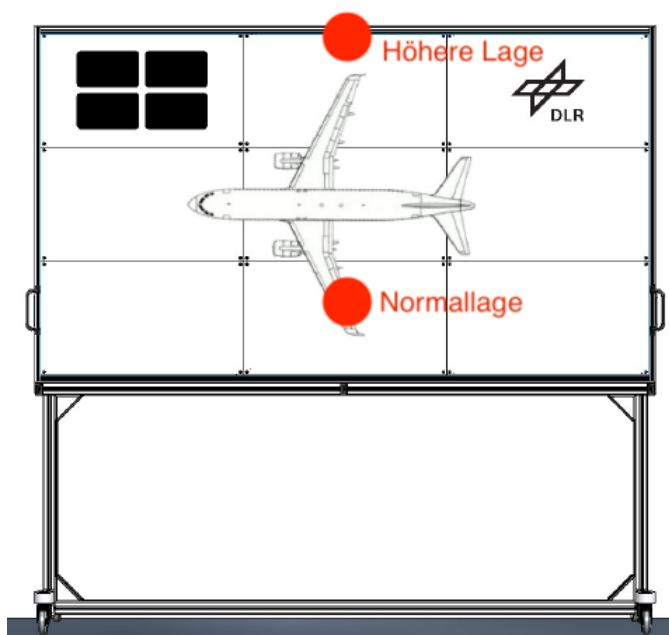


Abbildung 51: Kontaktpunkte beim Anlehnen

Das Schieben in der Normallage hat zur Folge, dass der Demonstrator durch die niedrige Reibung des Bodens ausweicht. Bei der höheren Lage kippt jedoch der Aufbau. Ab ungefähr 35 Grad Neigung kann der Aufbau einstürzen. Der Anlehntest mit dem Rücken hat ergeben, dass der Demonstrator nur mit Gewalt zum Sturz gebracht werden kann. Dadurch ist die Wahrscheinlichkeit des Einsturzes durch versehentliches Anlehnen minimal.

Erdung

Alle metallischen Teile sollen geerdet werden. Dadurch sollte bei versehentlicher Berührung des Phasenleiters eine Sicherung ausgelöst werden. Die elektrische Leitfähigkeit der Metallteile hängt hauptsächlich von der eloxierten Schicht auf dem Aluminium ab. Es wird nun einerseits geprüft, ob an allen Kontaktstellen die eloxierte Schicht ausreichend entfernt wurde, andererseits wird sichergestellt, dass zwischen allen metallischen Teilen und der PE Leitung ein ausreichend niedriger Widerstand liegt. Durch einen einfachen Kurzschlussstest bzw. eine Widerstandsmessung mit dem Multimeter lässt sich feststellen, ob die Erdung an dem äußeren Ende des Demonstrators eine Wirkung hat. Dafür wird das Messgerät in den Kurzschlussmodus umgestellt. Die erste Messspitze berührt den PE Anschluss des Hauptnetzteils. Die zweite Messspitze berührt ein Metallteil am entferntesten Ende des Demonstrators. Die Messung ergibt einige Milliohm, was von dem Multimeter bereits als eine leitende elektrische Verbindung erkannt wird.

Komponentenspannung

Die Komponentenspannung beträgt 24 VDC. Da sie unter 60 VDC ist, lässt sich auch im stromaktiven Zustand an den elektrischen Geräten, die für die Anwendung zuständig sind, arbeiten.

6.3 Allgemeiner Anwendungsfall

Auf Grundlage der Ergebnisse von 6.1 und 6.2 soll zusammengefasst werden, ob der Demonstrator als ein Forschungs- und Präsentationswerkzeug verwendet werden kann. Dabei sollen allgemeine Anwendungsschritte für die Arbeit mit dem Demonstrator formuliert werden, die als Grundlage für weitere Anwendungen dienen können. Um die allgemeinen

Anwendungsschritte zu formulieren, wird eine Beispielforschungsfrage auf Grundlage der Beispielanwendung umgesetzt.

Implementierung einer beispielhaften Forschungsfrage

Als erstes wird eine Frage formuliert, um einen Anwendungsfall implementieren zu können: *Kann der Demonstrator für wartungsrelevante Forschungszwecke eingesetzt werden?* Um die gestellte Frage umsetzen zu können, wird ein Anwendungsszenario definiert: Ein Wartungstechniker schließt ein Kalibrierungsgerät an den Sensor an und prüft, ob der unmittelbar am Sensor gemessene Wert dem Messwert im Cockpit entspricht. Der Luftfahrtforscher entwickelt ein neues Kalibrierungsgerät, um dem Wartungstechniker die Arbeit zu erleichtern.

Zunächst muss der Forscher bestehende Technologien und Prozesse studieren, Interviews mit Projektbeteiligten führen und sinnvolle Anforderungen formulieren. Als Beispiel werden in diesem Fall vier Flugmesssysteme dienen: Es sollten Flugsysteme für Bremstemperatur, Raddruck, statischen Druck und Gesamtlufttemperatur nachgebildet und in den Demonstrator eingebaut werden. Diese Systeme sollen miteinander kommunizieren und alle ihre Messwerte auf einem globalen Display anzeigen können. Der Forscher entscheidet sich dafür, das neue Kalibrierungswerkzeug vorab durch ein einfaches zusätzliches Display zu modellieren. Das Kalibrierungswerkzeug soll unmittelbar neben dem Sensor platziert werden.

Um festzustellen, welche Geräte für die Anwendung mindestens nachgebildet werden müssen, wird die technische Dokumentation studiert und sinnvolle Annahmen getroffen. Die Annahmen beeinflussen den Detailgrad der Nachbildung aber auch die Entwicklungsfreundlichkeit der Komponenten. Auf Grundlage der Annahmen und den Anforderungen werden Computer, Kommunikationseinheiten, Displays und Sensoren gewählt.

Die gewählten Komponenten werden sinnvoll gruppiert und in Anwendungsmodule aufgeteilt. Die Komponenten werden auf der Prozessebene über eine Hutschiene aufgesetzt. Innerhalb der Anwendungsmodule werden die elektronischen Komponenten mit Strom versorgt über eine lokale Spannungsverteilung mittels einer Reihenklemme. Weiterhin werden die elektronischen Komponenten verdrahtet bzw. über verpolungssichere Anschlüsse befestigt. Hierbei werden Komponenten, die für die Verarbeitung zuständig sind, auf der Prozessebene aufgesetzt. Die Komponenten, die für die Darstellung zuständig sind, werden auf der Präsentationsebene aufgesetzt. Über einen Verbindungsmanager kann die notwendige Kommunikationsstruktur umgesetzt werden. Die Softwareanwendung wird an eine standardisierte Kommunikationsschnittstelle angeschlossen, die über den Verbindungsmanager konfiguriert wird. Die Softwareanwendung wird entsprechend der Funktion der realen Systeme programmiert. Anschließend wird das System verifiziert und festgestellt, ob alle gestellten Anforderungen erfüllt werden.

Der Luftfahrtforscher hat jetzt die Möglichkeit, statt des zusätzlichen Displays neuartige Kalibrierungssysteme anzuschließen und sie miteinander zu vergleichen. Er kann seine Ergebnisse anderen Forschern vorstellen und auf dieser Grundlage Diskussionen führen. Es ist also möglich, den Demonstrator für wartungsrelevante Forschungszwecke einzusetzen. Der Mehraufwand ist auch geringer als wenn der Forscher für jeden neuen Anwendungsfall einen neuen Aufbau entwickeln würde.

Allgemeine Schritte für die Anwendung des Demonstrators

Die Beispielforschungsfrage gibt Aufschluss darüber, welche allgemeinen Schritte bei der Arbeit mit dem Demonstrator durchgeführt werden:

1. Formulierung einer Forschungsfrage
2. Definition eines Anwendungsszenarios
3. Studium der bestehenden luftfahrtrelevanten Technologien und Prozesse
4. Analyse der Anforderungen
5. Treffen sinnvoller Annahmen und Wahl der Geräte
6. Technische Umsetzung
7. Bewertung des Ergebnisses
8. Vorführung und Diskussion

Bei der technischen Umsetzung (6) müssen folgende Unterschritte durchgeführt werden:

1. Logische Gruppierung der Komponenten zu einem Anwendungsmodul
2. Umsetzung auf der Prozessebene
 - a. Mechanische Befestigung der Komponenten mit der Hutschiene
 - b. Anschluss der Komponenten an die globale Stromversorgung über Reihenklemmen
 - c. Verdrahtung der Komponenten über Schraubklemmen oder verpolungssichere Anschlüsse
3. Umsetzung auf der Präsentationsebene
 - a. Belegung einer Präsentationsfläche mit visuellen Elementen
 - b. Elektrische Verbindung zu der Prozessebene
4. Softwaretechnische Umsetzung
 - a. Anpassung des Verbindungsmanagers
 - b. Anschluss an die standardisierte Kommunikationsschnittstelle
 - c. Programmierung der Anwendungssoftware
5. Verifizierung der Funktion

7 Fazit

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst (Abschn. 7.1) und ein Ausblick auf die mögliche Weiterentwicklung gegeben (Abschn. 0).

7.1 Zusammenfassung

Das Ziel dieser Arbeit war die Entwicklung einer Flugzeugreferenzarchitektur in Form eines elektro-mechanischen Demonstratorstandes. Dieser sollte die bestehenden Flugsysteme nachbilden und als Grundlage für die Forschung an neuartigen Wartungssystemen in der Luftfahrtinstandhaltung dienen, so dass Forschungsthemen in Zukunft mit vermindertem Aufwand und geringeren Kosten umgesetzt und verifiziert werden können. Zur Eignungsprüfung des Demonstrators sollte zudem eine beispielhafte Anwendung implementiert werden, die vier Mess- und Anzeigesysteme des Avionics Compartments nachbildet.

In der Einleitung (Kap. 1) wurde die Verwendung der Flugzeugreferenzarchitektur motiviert, das Ziel definiert, der Rahmen der Bachelorarbeit beschrieben und die Vorgehensweise erläutert. In der Ausgangssituation (Kap. 2) wurden die Grundlagen der verteilten und eingebetteten Systeme sowie die in der Arbeit verwendete Middleware ROS beschrieben. Weiterhin wurde das Avionics Compartment und relevante Flugsysteme wie ARINC 429 Avionikbus, Electronic Instrument System, ADIRU, TPIS und BSCU dargestellt. Anschließend wurde ein Zusammenhang zwischen den beschriebenen Flugsystemen hergestellt.

Mit den Anforderungen (Kap. 3) begann der Hauptteil dieser Arbeit. Die Anforderungen wurden in zwei Gruppen aufgeteilt: Einerseits Anforderungen an einen Demonstratoraufbau, der eine präsentable mechanisch-elektronische Struktur darstellen soll und luftfahrtrelevante Komponenten trägt, andererseits Anforderungen an eine Beispielanwendung, die einen einfachen Messaufbau aus vier luftfahrtrelevanten Systemgrößen darstellt und dadurch Möglichkeiten des Demonstrators aufzeigt.

Die Anforderungen wurden mit den Projektbeteiligten durch Interviews identifiziert und einzeln analysiert, um die konkrete technische Realisierung zu ermöglichen. Anforderungen, die sich durch die Wahl konkreter technischer Geräte ergaben, wurden im Zuge der Konzeption (Kap. 4) konkretisiert.

Für den Demonstratoraufbau wurden folgende Eigenschaften identifiziert: Präsentierbarkeit, Erweiterbarkeit, Verfügbarkeit, Autarkie, Grenzmaße und Grenzgewicht sowie Personen- und Geräteschutz. Unter anderem wurde ermittelt, dass der Demonstrator optisch ansprechend sein sollte, einen Wechselstromanschluss benötigte und in logische Komponenten eingeteilt werden musste. Die Komponenten sollten schraublos und ohne Lötarbeiten befestigt bzw. verdrahtet werden können. Für die Beispielanwendung wurde eine Teilstruktur des Avionics Compartment identifiziert, die nachgebildet werden soll. Sie besteht aus EIS, ADIRU, BSCU und TPIS Flugsystemen. Für die Kommunikation sollte der ARINC 429 Bus nachgebildet werden, wobei nur die Verbindungs- und Adressierungsfunktion relevant ist. Mit diesen Systemen sollten Bremstemperatur, Raddruck, statischer Luftdruck und Gesamtlufttemperatur durch das verallgemeinerte Konzept der Data Modules gemessen werden. Diese sollten durch ein Netz von den genannten Flugsystemen weitergeleitet und auf einem System Display angezeigt werden.

Im Zuge eines nach Untersystemen und Unterkategorien strukturierten Auswahlprozesses wurden in der Konzeption (Kap. 4) anhand der in Kapitel 3 gestellten Anforderungen geeignete Systemkomponenten gewählt. Für die Beispielanwendung wurden unter anderem CAN-Bus Adapter, Raspberry PI's, ein Raspberry PI kompatibles Touchdisplay, Arduinos und relevante Sensoren selektiert. Als Middleware für die Kommunikation fiel die Wahl auf ROS. Für den Demonstratoraufbau wurden Aluprofile als Gerüst, Acrylglas als Abdeckungen und Hutschienen als Halterungssystem gewählt. Weiterhin wurde ein Schaltnetzteil für die Wechselstromwandlung gewählt, die über eine gemischte Netzverteilungsform Strom an die einzelnen Anwendungsmodule verteilt. Die Stromgrenze für das Schaltnetzteil wurde zusammen mit dem entsprechenden Kabelquerschnitt für die Stromverteilung auf Grundlage des Stromverbrauchs der Raspberry PI's ausgesucht. Innerhalb der Anwendungsmodule wurde die Spannung durch DC-DC Wandler für die Endverbraucher angepasst.

Die Entwicklung (Kap. 5) basiert auf dem Strukturkonzept aus der Konzeption (Kap. 4). Zuerst wurde der Demonstratoraufbau geplant und umgesetzt, danach die Beispielanwendung realisiert. Für die Handhabung der hohen Komplexität des Aufbaus wurden Funktionsebenen und Anwendungsmodule definiert. Die Anwendungsmodule teilen die Komponenten in unabhängige logische Einheiten, die Funktionsebenen in Präsentations-, Prozess- und Serviceebenen auf.

Bei dem Demonstratoraufbau wurde zuerst der mechanische Aufbau umgesetzt. Das Gerüst wurde zusammengebaut und die Hutschienen und Abdeckungen daran befestigt. Bei den Abdeckungen wurden Löcher geschnitten, um Komponenten für die Stromversorgung und Visualisierung integrieren zu können. Die Versorgungskomponenten wurden auf die Hutschiene aufgesetzt und verdrahtet. Es wurden Stromversorgungskabel zu den Anwendungseinheiten verlegt und mit der lokalen Reihenklemme verbunden. Alle Metallteile wurden durch den PE Leiter geerdet.

Für die geforderten Flugsysteme wurde jeweils ein Anwendungsmodul aus Komponenten wie Raspberry PI's, Arduinos, OLED-Display, Touch Displays und Sensoren zusammengesetzt und entsprechend dem im Kap. 3 vorgestellten Plan verbunden. Sensoren wie auch lokale Displays wurden auf spezielle Platten aufgesetzt und in der Präsentationsebene integriert. Auf Prozessebene wurden sie mit Arduinos verknüpft, die dann wiederum an die PI's angeschlossen wurden. Die PI's wurden über die CAN-USB Adapter und spezielle Leitungen miteinander verbunden. Die DC-DC Wandler wurden angepasst, um den PI's die benötigte Spannung zur Verfügung zu stellen.

Für die programm-technische Kommunikationsverbindung wurde ROS und socketCAN verwendet. Es wurde ein Verbindungsmanager entwickelt, der die gesamte Kommunikation konfiguriert und verkapselt. Die entwickelte Architektur bedient sich dabei einer ARINC-ähnlichen Verbindungs- und Adressierungsart. Die Anwendungen der jeweiligen Computer wurden unter Verwendung des Verbindungsmanagers entsprechend ihrer Funktion und dem geforderten Verbindungsplan in der ROS Umgebung programmiert. Für die automatische Initialisierung der CAN-Bus Adapter im Linux Betriebssystem wurden spezielle Skripte geschrieben und udev Regeln aufgestellt. Für das automatische Starten der ROS Programme ohne grafische Oberfläche wurde Robot-Upstart verwendet. Für das automatische Starten von Programmen mit grafischer Oberfläche wurde das „Startup Application“ Werkzeug des Ubuntu Betriebssystems verwendet.

Im Test (Kap. 6) wurde die tatsächliche Funktion des Demonstrators in drei Etappen ermittelt. Im Zuge der Funktionsverifikation der Beispielanwendung wurden grundsätzliche Tests an den Sensoren und Displays durchgeführt und damit Normalwerte und Anzeigen erfolgreich verifiziert.

Weiterhin wurde ermittelt, dass die Latenzzeit das Aktualisierungsintervall nicht übersteigt. Die Verdrahtung und Verbindung in der aktuellen ARINC 429 Nachbildung hat sich als unkompliziert erwiesen, wenngleich reale ARINC 429 Geräte nicht direkt angeschlossen werden können.

Bei der Verifikation der Demonstrortechnik wurden alle geforderten Eigenschaften jeweils einzeln und mit der Beispielanwendung als Abstoßpunkt untersucht. Unter anderem wurde festgestellt, dass der Demonstrator einen guten optischen Eindruck vermittelt und leicht erweitert werden kann. Er benötigt durchschnittlich 77 Sekunden für das Hochfahren der Computer und Programme. Es können insgesamt 40 Raspberry PI's verbaut werden. Von dem aktuellen Hauptnetzteil können insgesamt 25 Computer mit zusätzlicher Elektronik versorgt werden. Der Demonstrator bleibt beim versehentlichen Anlehnen kippstabil und es kann auch bei aktiver Komponentenspannung im Inneren des Demonstrators gearbeitet werden.

Abschließend wurde anhand einer Beispielforschungsfrage gezeigt, wie genau der Demonstrator in den Forschungsprozess integriert werden kann. Für die Konkretisierung der Forschungsfrage wurde ein rollenbasiertes Anwendungsszenario mit einem Techniker, einem Luftfahrtforscher und einem Kalibrierungswerkzeug entworfen und die generellen Schritte für die Anwendung des Demonstrators aufgeführt. Die Erkenntnisse aus der Beispielanwendung ließen zusammen mit dem vorgestellten Szenario den Schluss zu, dass solche Anwendungsfälle mit dem Demonstrator gut umsetzbar sind und die aktuelle Demonstrator-Infrastruktur für den Einsatz als Forschungswerkzeug in der Luftfahrt durchaus geeignet ist.

7.2 Ausblick

Die aktuelle Version des Demonstrators kann bereits jetzt für viele Anwendungen verwendet werden. Um die Produktivität jedoch noch weiter zu erhöhen, könnten in Zukunft eine Reihe von Verbesserungen durchgeführt werden.

So sind die Präsentations- und Prozessebenen von Komponenten in der aktuellen Umsetzung voneinander unabhängig. Für eine höhere Modularität und Effizienz könnten diese Ebenen in einem einzelnen elektrischen und programmtechnischen Anwendungsmodulkonzept vereint werden. Zudem könnten standardisierte Schnittstellen entwickelt werden, so dass jedes Anwendungsmodul in wenigen Handgriffen als ein Ganzes an die Infrastruktur des Demonstrators angeschlossen werden könnte. Eine hohe Anzahl an Rechnerkomponenten samt einfachem und strukturiertem Zugriff auf entsprechende Computer, Betriebssysteme und Mikrocontroller könnte an dieser Stelle durch angepasste Softwaremanagement- und Softwareentwicklungssysteme ermöglicht werden.

Der nächste Schritt wäre die Integration realer ARINC 429 Konnektivität, z. B. über Adapter, die über einen USB-Port an die Avionics Computer anschließbar sind. Mithilfe dieser Integration könnte der Demonstrator mit realen Flugsystemen wie z. B. einem Flugschreiber verbunden werden. Dadurch eröffnen sich für den Demonstrator mehr Forschungsfragen, bei denen er angewendet werden könnte. Statt der bereitgestellten Modellimplementierungen könnte im Demonstrator an geeigneten Stellen zum Beispiel eine spezielle luftfahrtzertifizierte Hardware eingesetzt werden. Einen echten Mehrwert würde umgekehrt auch die Möglichkeit bieten, die Demonstrator-Hardware in bereits bestehende Umgebungen der Luftfahrttechnik zu integrieren.

Als Weiterentwicklung der Autarkie könnte ein Transportkonzept ausgearbeitet und implementiert werden. Um den Demonstrator leicht innerhalb einer lokalen Einrichtung zu verschieben, könnte er mit passiven Rollen ausgestattet werden. Auch der Transport zu einem

Ausstellungsort in einem Fahrzeug wäre wünschenswert. Dafür müsste eine Konstruktionsmodifikation in Richtung eines modularen, teilbaren Aufbaus des Gesamtgerüsts erfolgen.

In ferner Zukunft könnte der Demonstrator serienreif gefertigt und universalisiert werden, um jegliche Anwendungsfälle mit gängiger wie auch experimenteller Elektronik in der Luftfahrtforschung realisieren zu können. Man könnte vorgefertigte Bausteine auf Basis von ARM Einplatinencomputern, Mikrokontrollern oder FPGAs entwickeln, so dass jeder Anwendungsfall in sehr kurzer Zeit umgesetzt werden würde. Die Entwicklungsumgebung sollte hierbei soweit spezialisiert und universalisiert werden, dass der Forscher so gut wie keinen technisch begründeten Mehraufwand mehr bewältigen muss und sich allein auf die Entwicklung und Forschung an luftfahrtbezogenen Anwendungen konzentrieren kann.

Quellenverzeichnis

- [Airb06] AIRBUS S.A.S.: A320 - ATA 31 - SINGLE AISLE TECHNICAL TRAINING MANUAL MAINTENANCE COURSE - M35 LINE MECHANICS (CFM56-5B/ME) INDICATING/RECORDING SYSTEMS, 2006. — Benutzerhandbuch
- [Airb12] AIRBUS: A320 - ATA 31 - Aircraft Maintenance Manual, 2012. — Benutzerhandbuch
- [Airb96a] AIRBUS INDUSTRIE: A320 - ATA 32 - Landing Gear, 1996. — Benutzerhandbuch
- [Airb96b] AIRBUS INDUSTRIE: A320 - ATA 34 - Navigation, 1996. — Benutzerhandbuch
- [Arin04] ARINC: MARK 33 DIGITAL INFORMATION TRANSFER SYSTEM (DITS) PART 1 FUNCTIONAL DESCRIPTION, ELECTRICAL INTERFACE, LABEL ASSIGNMENTS AND WORD FORMATS, 2004. — Datenblatt
- [Aust08] AUSTRIAN TECHNIK: A320 - ATA 31 - Indication/Recording Systems, 2008. — Benutzerhandbuch
- [Bapa94] BAPAT, SUBOTH: Object-Oriented Networks: Models for Architecture, Operations, and Management : Englewood Cliffs, 1994
- [Bghm12] BGHM: BG-Information - Elektrofachkräfte - BGI 548, 2012. — Datenblatt
- [Coll19] COLLINS, MICHAEL: Computational Graphs, and Backpropagation, Course Notes on NLP. Columbia University, 2019. — Vorlesungsskript
- [Cond00] CONDOR ENGINEERING: ARINC Protocol Tutorial, 2000. — Benutzerhandbuch
- [Cpu20] CPU, WORLD: Motorola 68020 (MC68020) microprocessor family. URL <http://www.cpu-world.com/CPUs/68020/index.html>. - abgerufen am 2020-05-24
- [Dani17] DANIEL: Deep Learning From Scratch I: Computational Graphs. URL <https://www.deepideas.net/deep-learning-from-scratch-i-computational-graphs/>. - abgerufen am 2020-05-24
- [DiHa03] DIEKMANN, THOMAS ; HAGENHOFF, SVENJA: Verteilte Systeme: State of Art (Nr. 1). Institut für Wirtschaftsinformatik, Georg-August-Universität Göttingen, 2003. — Arbeitsbericht
- [Ecam00] ECAM Images. URL <https://digilander.libero.it/miovolo/ecam.htm>. - abgerufen am 2020-04-09

- [Flüh10] FLÜHR, HOLGER: Avionik und Flugsicherungstechnik: Einführung in Kommunikationstechnik, Navigation, Surveillance. Berlin Heidelberg : Springer-Verlag, 2010 — ISBN 978-3-642-01612-7
- [Foot00] FOOTE, TULLY: ROS Concept. URL <http://wiki.ros.org/de/ROS/Concepts>. - abgerufen am 2020-05-24
- [John15] JOHN ANTHONY, RICHARD: Systems Programming: Designing and Developing Distributed Applications : Morgan Kaufmann Publishers, 2015
- [Kabe00] BEKON KABEL: Auszug aus den VDE-Vorschriften 0100, Teil 430, DIN 57100, Teil 430. URL <http://bekon.de/pdf/pdf/vdevor1.pdf>. - abgerufen am 2020-05-24
- [LaFe12] LAMERSDORF, W. ; FEDERRATH, H.: Verteilte Systeme und Informationssicherheit (VIS). Universität Hamburg, Hamburg, 2012. — Vorlesungsskript
- [Meme00] ME-MESSYSTEME GMBH: Grundlagen zum CAN Bus. URL <https://www.me-systeme.de/de/technik-zuerst/elektronik/can-bus-grundlagen>. - abgerufen am 2020-04-11
- [Ndim00] ND Image. URL http://www.a320dp.com/A320_DP/flight-instruments/sys-8.1.0.html. - abgerufen am 2020-04-09
- [Neum14] NEUMEYER, DAVID: The Oxford Handbook of Film Music Studies : OUP USA, 2014
- [Ngai09] NGAI, EDITH: Distributed Information Systems 1DT057. Uppsala University, Uppsala, Sweden, 2009. — Vorlesungsskript
- [Odin17] ODINETS, TATIANA: Sizing of essential SHM components of IVHM architecture and computing the relevant costs via scalable simulation. Hamburg, Institut für Lufttransportsysteme and Technische Universität Hamburg-Harburg, Masterarbeit, 2017
- [PaHe00] PAHLKE, STEPHAN ; HERZFELD, SVEN: CAN-Bus: Fehlererzeugung und Analyse. Hannover, Fachhochschule Hannover, Diplomarbeit, 2000
- [Pfdi00] PFD Image. URL http://www.oaviao.com/pesquisa_codigo_empresas/manuais_voo/performance_airbus/images/11.gif. - abgerufen am 2020-04-23
- [Ros00] ROS: ROS Core Components. URL <https://www.ros.org/core-components/>. - abgerufen am 2020-05-25
- [Siem12] SIEMERS, CHRISTIAN: Embedded Systems Engineering. Technische Universität Clausthal, Clausthal, 2012. — Vorlesungsskript
- [Wenz18] WENZEL, ANDRE: Prozessleittechnik und Bussysteme (SS18). HAW Hamburg, Hamburg, 2018. — Vorlesungsskript

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____